Contributions to Lane Marking Based Localization for Intelligent Vehicles

Wenjie Lu

University of Paris-Sud

DOCTORAL SCHOOL 422:
SCIENCE AND INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS SYSTEMS

Laboratory: Institute of Fundamental Electronics, UMR CNRS 8622

DOCTORAL THESIS

INFORMATION SCIENCES AND COMMUNICATION

by

Wenjie LU

Contributions to Lane Marking Based Localization for Intelligent Vehicles

Date of defense: 09/02/2015

Thesis Committee:

Reviewers:	Michel Devy	Research Director	(LAAS-CNRS)
	Fawzi Nashashibi	Research Director	(RITS-INRIA)
Examiners:	Vincent Frémont	Associated Professor	(Heudiasyc UMR UTC)
Supervisors:	Roger Reynaud	Professor	(University Paris-Sud)
	Sergio A. Rodríguez F.	Associated Professor	(University Paris-Sud)
	Emmanuel Seignez	Associated Professor	(University Paris-Sud)

UNIVERSITÉ PARIS-SUD

ÉCOLE DOCTORALE 422:
SCIENCES ET TECHNOLOGIES DE L'INFORMATION DES
TÉLÉCOMMUNICATIONS ET DES SYSTÈMES

Laboratoire : Institut d'Electronique Fondamentale, Unité Mixte de Recherche CNRS 8622

THÈSE DE DOCTORAT

SCIENCES DE L'INFORMATION ET DE LA COMMUNICATION

par

Wenjie LU

Contribution à la Localisation de Véhicules Intelligents à Partir de Marquage Routier

Date de soutenance: 09/02/2015

Composition du jury:

Rapporteurs:	Michel Devy	Directeur de Recherche	LAAS-CNRS
	Fawzi Nashashibi	Directeur de Recherche	RITS-INRIA
Examinateurs:	Vincent Frémont	Maître de Conférence HDR	Heudiasyc UMR UTC
Directeurs de thèse:	Roger Reynaud	Professeur des Universités	Université Paris-Sud
	Sergio A. Rodríguez F.	Maître de Conférences	Université Paris-Sud
	Emmanuel Seignez	Maître de Conférences	Université Paris-Sud

Abstract

Autonomous Vehicles (AV) applications and Advanced Driving Assistance Systems (ADAS) relay in scene understanding processes allowing high level systems to carry out decision marking. For such systems, the localization of a vehicle evolving in a structured dynamic environment constitutes a complex problem of crucial importance. Our research addresses scene structure detection, localization and error modeling. Taking into account the large functional spectrum of vision systems, the accessibility of Open Geographical Information Systems (GIS) and the widely presence of Global Positioning Systems (GPS) onboard vehicles, we study the performance and the reliability of a vehicle localization method combining such information sources. Monocular vision–based lane marking detection provides key information about the scene structure. Using an enhanced multi-kernel framework with hierarchical weights, the proposed parametric method performs, in real time, the detection and tracking of the ego-lane marking. A self-assessment indicator quantifies the confidence of this information source. We conduct our investigations in a localization system which tightly couples GPS, GIS and lane makings in the probabilistic framework of Particle Filter (PF). To this end, it is proposed the use of lane markings not only during the map-matching process but also to model the expected ego-vehicle motion. The reliability of the localization system, in presence of unusual errors from the different information sources, is enhanced by taking into account different confidence indicators. Such a mechanism is later employed to identify error sources. This research concludes with an experimental validation in real driving situations of the proposed methods. They were tested and its performance was quantified using an experimental vehicle and publicly available datasets.

Résumé

Les applications pour véhicules autonomes et les systèmes d'aide avancée à la conduite (Advanced Driving Assistance Systems - ADAS) mettent en œuvre des processus permettant à des systèmes haut niveau de réaliser une prise de décision. Pour de tels systèmes, la connaissance du positionnement précis (ou localisation) du véhicule dans son environnement est un pré-requis nécessaire. Cette thèse s'intéresse à la détection de la structure de scène, au processus de localisation ainsi qu'à la modélisation d'erreurs. A partir d'un large spectre fonctionnel de systèmes de vision, de l'accessibilité d'un système de cartographie ouvert (Open Geographical Information Systems - GIS) et de la large diffusion des systèmes de positionnement dans les véhicules (Global Positioning System - GPS), cette thèse étudie la performance et la fiabilité d'une méthode de localisation utilisant ces différentes sources. La détection de marquage sur la route réalisée par caméra monoculaire est le point de départ permettant de connaître la structure de la scène. En utilisant, une détection multi-noyau avec pondération hiérarchique, la méthode paramétrique proposée effectue la détection et le suivi des marquages sur la voie du véhicule en temps réel. La confiance en cette source d'information a été quantifiée par un indicateur de vraisemblance. Nous proposons ensuite un système de localisation qui fusionne des informations de positionnement (GPS), la carte (GIS) et les marquages détectés précédemment dans un cadre probabiliste basé sur un filtre particulaire. Pour ce faire, nous proposons d'utiliser les marquages détectés non seulement dans l'étape de mise en correspondance des cartes mais aussi dans la modélisation de la trajectoire attendue du véhicule. La fiabilité du système de localisation, en présence d'erreurs inhabituelles dans les différentes sources d'information, est améliorée par la prise en compte de différents indicateurs de confiance. Ce mécanisme est par la suite utilisé pour identifier les sources d'erreur. Cette thèse se conclut par une validation expérimentale des méthodes proposées dans des situations réelles de conduite. Leurs performances ont été quantifiées en utilisant un véhicule expérimental et des données en libre accès sur internet.

Contents

Acronyms

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
AV	Autonomous Vehicles
BEV	Bird's Eye View
DGNSS	Differential GNSS
DR	Dead Reckoning
ECEF	Earth-Centered Earth-Fixed (coordinate)
EKF	Extended Kalman Filter
ENU	East-North-Up (coordinate)
ESS	Effective Sample Size
FA	False Alarms
FHD	Finlayson-Hordley-Drew
FN	False Negative
FP	False Positive
GIS	Geographical Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HIS	Hue, Saturation, Intensity
HOG	Histogram of Oriented Gradients
HSL	Hue, Saturation, Lightness
HSV	Hue, Saturation, Value
IMU	Inertial Measurement Units
INS	Inertial navigation system
IPM	Inverse Perspective Mapping
IV	Intelligent Vehicle
LBS	Location Based Service
LDW	Lane Departure Warning system
LIDAR	LIght Detection And Ranging
MAE	Mean Absolute Error
LSM	Least Square Method

MAE	Mean Absolute Error
MKE	Multi-Kernel Estimation
MSE	Mean Squared Error
NMEA	National Marine Electronics Association
OS	Ordnance Survey
OSM	OpenStreetMap
PDF	Probability Density Function
PLD	Predictive Lane Detection
RANSAC	RANdom SAmple Consensus
RGB	Red, Green, Blue
RMSE	Root-Mean-Square Error
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
SDRE	State-Dependent Riccati Equation
SHT	Statistical Hough Transform
SLAM	Simultaneous Localization And Mapping
SLR	Scanning Laser Radar
SIR	Sampling Importance Resampling
SIS	Sequential Importance Sampling
TN	True Negative
TP	True Positive
UM	Urban Marked
UMM	Urban Marked Multi-lane
UU	Urban Unmarked
WGS-84	World Geodetic System 1984

List of Figures

Chapter 1

General Introduction

Vehicle localization stands for the global coordinates position coordinate of the rover, which is a key component of Intelligent Vehicle (IV), a prerequisite for Advanced Driver Assistance System (ADASs), and the foundation of Location Based Services (LBSs). After a precise GPS navigation became open to the public in 2000, on-vehicle localization and navigation technologies boomed in both industrial field and scientific field. A number of great achievements in vehicle localization have been reached during the past decade. For instance, mature products have been widely used in civilian vehicles. Some representative systems are presented in Fig. 1.1. These products can be a separate module as Fig. 1.1b-c attached to the vehicle, or embedded into the driver panel by manufacturers. In scientific area, the research on vehicle localization is one more step forward than commercial products. Some aim at improving the accuracy of localization via GNSS receivers only, some employ multi-sensor to eliminate the noise of GPS signals, some fuse Geographical Information System (GIS) and GPS together to obtain precise vehicle positions, while some focus on Simultaneous Localization And Mapping (SLAM).

Fig. 1.2 illustrates examples of the experimental intelligent vehicle platform. The AMADEO vehicle platform is developed by Autonomous Systems Laboratory of Uni-

(a) *(b)* *(c)*

Figure 1.1: *Examples of vehicle navigation products. (a) an embedded navigation system on a RENAULT® vehicle; (b) a Tomtom® navigation product; (c) a Becker® map pilot box.*

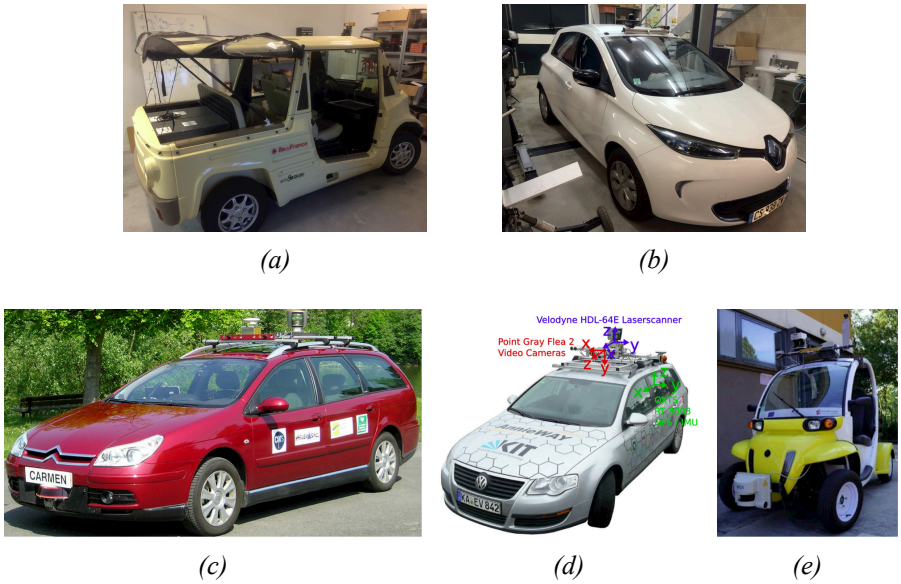


Figure 1.2: *Examples of intelligent vehicle platform. (a) AMADEO experimental vehicle of Autonomous Systems, Université Paris-Sud; (b) Concept car ZOE of Project RTRA DIGITEO Roboteo Handler, Université Paris-Sud; (c) test vehicle from Heudiasyc Laboratory, Université de Technologie de Compiègne; (d) vehicle platform from Karlsruhe Institute of Technology (KIT); (e) equipped vehicle of l'Institut de Recherche sur les Transports, l'Énergie et la Société (IRTES);*

versité Paris-Sud, shown in Fig. 1.2a. This experimental vehicle consists of an autonomous vehicle equipped sensors (cameras, LIDAR, odometers), actuators (steering, propulsion, brake, brake drive) and embedded computer systems (computer, communication nodes, CAN bus). Fig. 1.2b depicts a RENAULT® ZOE, which is a concept car for Project RTRA DEGITEO Roboteo Handler. The test car in Fig. 1.2c comes from Heudiasyc Laboratory of the Université de Technologie de Compiègne. The vehicle was equipped with a NovAtel RTK-GPS receiver coupled with a SPAN-CPT IMU running at 10Hz. The system received RTCM 3.0 corrections through a 3G connection from a GPS base station Septentrio PolaRx2e equipped with a Zephyr Geodetic antenna. Fig. 1.2d is an experimental platform developed by Karlsruhe Institute of Technology (KIT). The mounted sensors include one OXTS RT 3003 Inertial Navigation System (GPS/IMU), a laserscanner of Velodyne HDL-64E, two gray scale cameras, two color cameras, and four NT59-917 varifocal lenses. Fig. 1.2e is an electrical GEM car[1] based test bed from Laboratoire Systèmes et Transports of l'Institut de Recherche sur les Transports, l'Énergie et la Société (IRTES), equipped with ProFlex 500 Magellan RTK-GPS receiver, Bumblebee XB3 system stereoscopic system, and horizontal SICK LMS221 laser range finder. These experimental vehicles are suitable for tests on ADASs, including vehicle localization.

Although vehicle positioning has been studied for years, difficulties still remain in this domain. Commercial products have provided convenient and mature navigation

[1] http://www.gemcar.com/

functions based on GPS only. But the positioning ability of these products decreases in urban environment, because of satellite signal outages and GPS signal multi-path propagation caused by high buildings. This problem has been alleviated via multi-sensor based data fusion methods. For instance, Inertial Navigation System (INS) can measure velocities, rotations and other vehicle dynamics to alleviate GPS noises. High-accuracy maps with road features are combined with cameras to assist vehicle positioning. Data fusion algorithms match the road features from vision to those in the map database, so as to adjust the GPS signals. However, these methods are unable to apply in common vehicles. Sensors such as INS have large size and high cost. High accuracy maps with precise road features have created manually, so that these maps are expensive and restricted in particular regions. Therefore, the problem is stated: **how to develop a localization system with low cost, which can be suitable for common cars in urban environment?**

To this end, we propose a marking feature based vehicle localization method using low-cost sensors and open source map, with a probabilistic error modeling. The idea is to extract marking features from vision and match these features with lane marking topology obtained from map database. We can observe the system structure in Fig. 1.3. The top part is sensor input, included only two different types of low-cost sensors, a monocular camera and an ordinary GPS receiver. These sensors are easier to implement on vehicles and much cheaper than IMU. GPS is the fundamental sensor of vehicle positioning, camera is used to capture road scene photos, from which the lane markings can be extracted. The left part is open source map database. Lane marking features are derived from map information such as road type parameter and number of lanes parameter. The box on the right side in Fig. 1.3 is the system organization. At first, lane markings are detected using a Multi-Kernel Estimation (MKE) method from an on-vehicle camera. A particle filter is implemented to estimate the vehicle position with respect to the detected markings. Then map-based markings are constructed according to an open source map database. Vision-based markings and map-based markings are fused through MKE to obtain the adjusted vehicle fix. In order to further improve the localization performance, a probabilistic error model is employed to identify the possible errors. Finally, the vehicle position with error modeling is output in the bottom part.

To meet low cost requirement, a camera and a GPS receiver is implemented in the vehicle, open source maps, which can be downloaded for free, are also employed into the proposed system. To satisfy the accuracy of localization, three steps are exploited: 1) a lane marking shape based particle filter tunes the GPS position; 2) MKE based vehicle localization improves vehicle position via fusing both markings from image processing and markings produced by map database; 3) a probabilistic error modeling is set to detect potential errors in localization. Therefore, a vehicle localization method

with low-cost sensors and open source map database is completed.

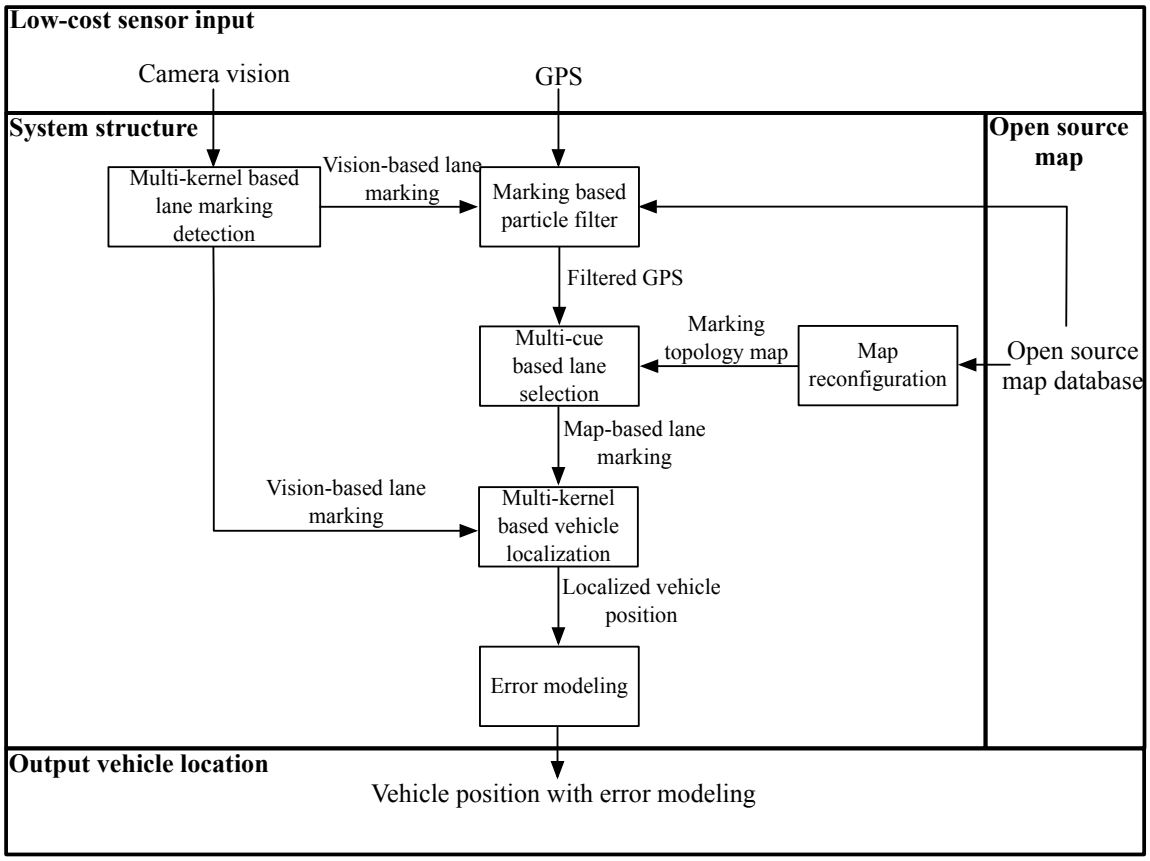


Figure 1.3: *Project structure represented in this thesis. Multi-kernel based lane marking detection will be introduced in Chapter 2, map reconfiguration, marking based particle filter, multi-cue based lane selection and vehicle localization will be represented in Chapter 3, and algorithms of error modeling will be proposed in Chapter 4.*

The outline of the dissertation is arranged as follows. Lane marking models are extracted in image processing state, and fitted using MKE method in Chapter 2. In Chapter 3, lane marking topology is constructed, a marking based particle filter is implemented, current lane is selected by multi cues, and the vehicle position is located using an MKE method. The possible errors are detected through a probabilistic error modeling method in Chapter 4. Chapter 5 presents experimental results on a test vehicle.

Chapter 2

Multi-kernel Estimation based Lane Marking Detection

Contents

2.1 Introduction

This chapter presents an enhanced lane marking detection approach intended for low-level perception. The work of this part has been published in Lu et al. (2014a, 2013).

A pair of lane markings of current lane in structured environments is produced as an output, which is used as key information for lane marking based vehicle localization. This method relies on a multi-kernel detection framework with hierarchical weights. First, the detection strategy performs in BEV space and starts with an image filtering using a cell-based blob method. Then, marking parameters are optimized following a parabolic model. Finally, a self-assessment process provides an integrity indicator to improve the output performance of detection results.

In lane detection part, the camera images are captured in perspective space, which will be projected into BEV space for image processing. The image processing stage and marking detection stage are implemented in BEV space. The output marking results are projected back to perspective space.

On-vehicle lane marking detection plays an important role in Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles (AV). In this context, this function constitutes a key part of low-level perception, which is intended to help high-level system by estimating the shape and the localization of the lane markings. Such information can be exploited within more complex systems, for instance, Automatic Cruise Control (ACC) Kim and Yi (2013), drowsiness detection systems Li et al. (2012), scene understanding Rodríguez F. et al. (2014), and vehicle localization systems Schreiber et al. (2013).

The difficulties facing in lane detection process are divided into two contradicts, one is the conflict between high process cost and real time requirement, the other is various road environments and high accuracy required by high level systems. Very few assumptions on scene appearance hold true for longer than $5 - 10$ seconds Lombardi et al. (2005). However, too many assumptions on detection algorithm may decrease the detection precision. For instance, straight line marking model brings conveniences in calculation, but performs poor results at curves. Unpredictable and numerous different road cases also raise difficulties in marking detection. On the other side, ADAS should reach high reliability and low error rates. For example, the acceptable false alarm of a Lane Departure Warning system (LDW) should be very row Burzio et al. (2010); Barickman et al. (2007).

Searching for balances of these two contradicts, this subject has been widely studied for decades and recent remarkable progresses have been summarized in Hillel et al. (2014); Yenikaya et al. (2013). In this state-of-the-art, the main difficulties in lane detection are enumerated at first. Then most of existing lane detection approaches follow a common strategy composed of the following stages: information-adopting sensors, preprocessing

stage, feature extraction stage, and model fitting stage. In feature extraction part, the potential features of lane markings are classified, and the extraction methods are analyzed. In model fitting part, different marking models are introduced before regression analysis for model estimation. To clarify the architecture of the lane marking detection system, a flowsheet is introduced to demonstrate all the stages announced as Fig. 2.1.

2.1.1 Road Scenarios

Different road scenarios affect lane marking detection results greatly. Coping with diverse road scenarios in detection algorithm is a key factor to reach high accuracy. A relatively simple Hough transform-based algorithm solves roughly 90% of highway cases, if the road markings are clearly marked from input sources Borkar et al. (2009a). The common complex scenarios for lane marking detection can be summarized into two categories: the internal disturbs, which are raised by the lane markings themselves, and the external disturbs, which are caused by extrinsic reasons. Fig. 2.2 depicts examples of road diversity.

Internal disturbs. The difficulties brought by lane markings are classified into internal disturbs. Lane markings are normally 0.1 meter wide with yellow or white painted colors. But numerous variants exist. For instance, marking is too wide, the color is contaminated (Fig. 2.2b) or fades (Fig. 2.2a). With regard to the standard lane width of $2.5 - 3.25$ meters, in some scenarios, a pair of lane markings in a lane in too narrow on too wide, as observed in Fig. 2.2f. In some scenes such as Fig. 2.2e, the marking of one side is missing, while in some conditions as illustrated in Fig. 2.2c, arrows, zebra lines and other markers exist on the road.

External disturbs. The problems in this category are caused by several sources. Weather conditions (i.e. snow, rain, fog, etc.) take vision difficulties into lane detection, as shown in Fig. 2.2d. Besides extreme climates, lighting conditions like shadow, blaze, and night obviously impact the vision quality, as given in Fig. 2.2e and Fig. 2.2f. Additionally, the lane markings are sometimes occluded by barriers or other vehicles on the road. And road scars on highway surfaces can be mistaken for markings, presented in Fig. 2.2f.

2.1.2 Sensor Types

Different sensors and perception modalities are used within the state-of-the-art. The advantages and disadvantages of enumerated sensors are discussed.

Laser-based sensors. LIght Detection And Ranging (LIDAR) is a remote sensing technology that measures distance by illuminating a target with a laser and analyzing the reflected light. This property ensures the resistance to nature lights (i.e. strong lights, shadows) and is able to report 3D structure and the reflected intensity of an

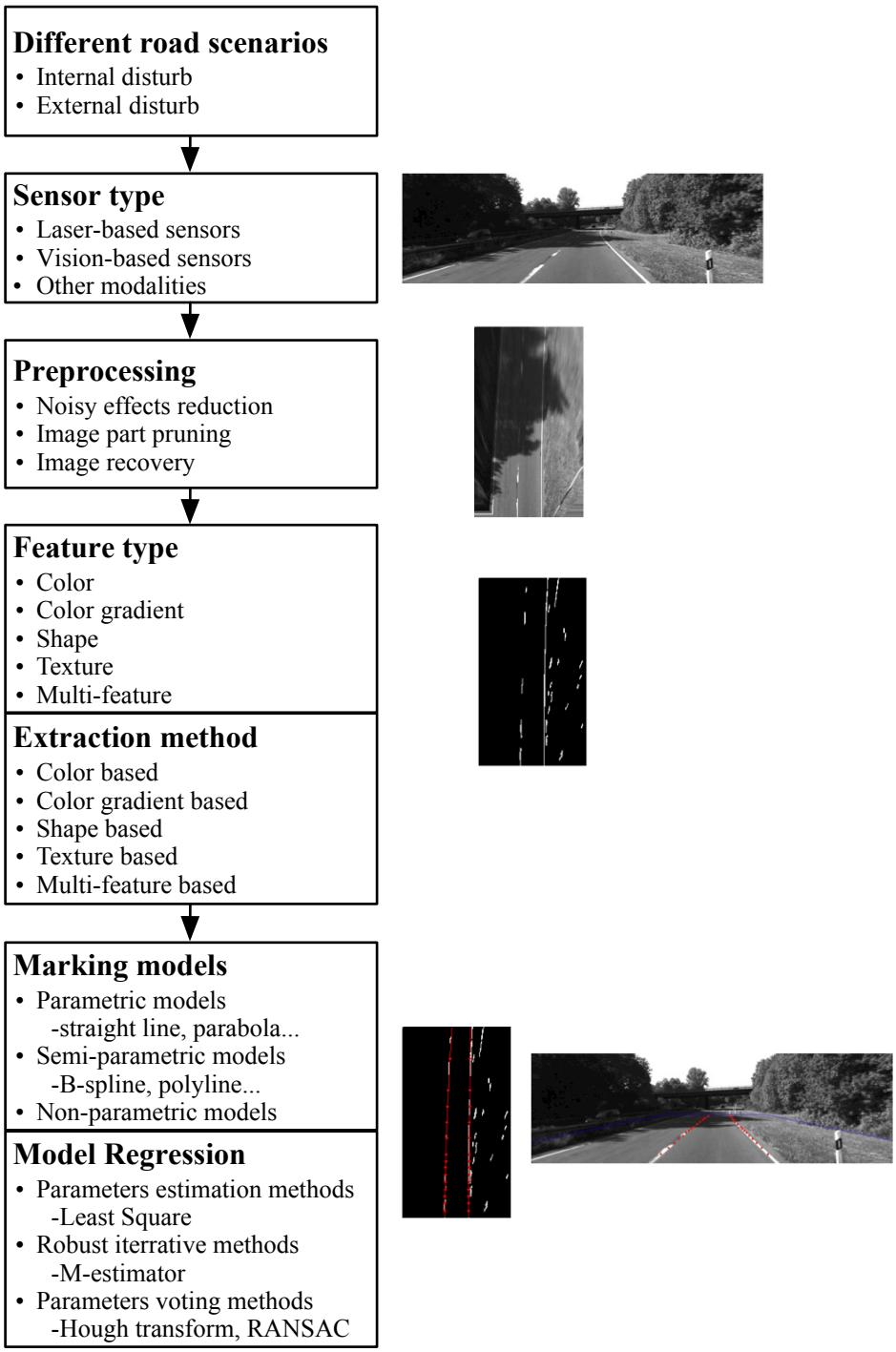


Figure 2.1: *State-of-the-art on lane marking detection system. A standard lane marking detection system is classified into five parts: road scenarios, sensor types, preprocessing stages, feature extraction, and model fitting.*

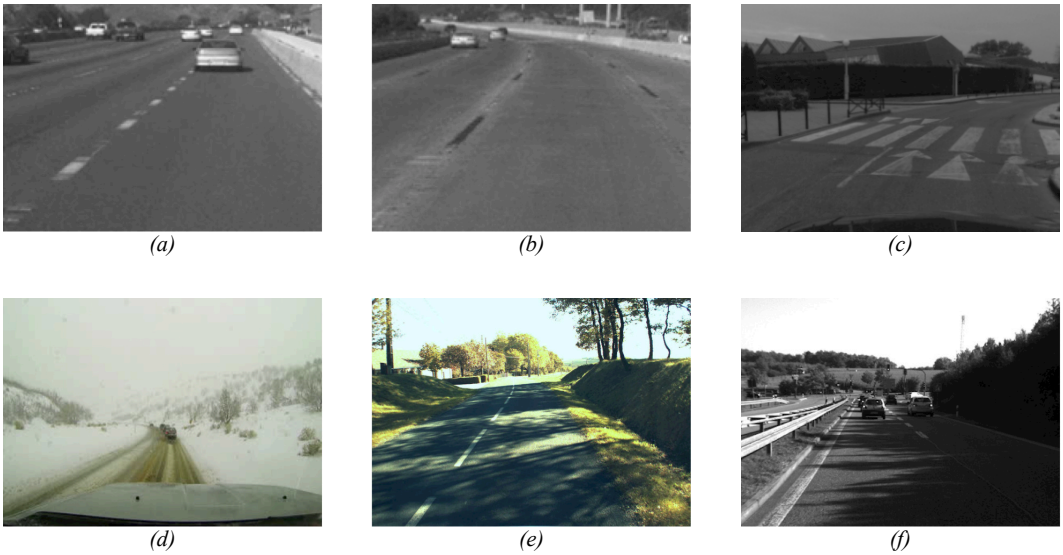


Figure 2.2: *Examples of road diversity. (a) faded markings; (b) contaminated markings; (c) disturb markers; (d) snow; (e) shadow+missing right marking; (f) shadow+too wide lane width+road scar.*

object, leading to measure the roughness, curbs and berms on the road. What's more, road slope and rover pitch angle can be derived by LIDAR Labayrade et al. (2006); Nieto et al. (2008). Thus LIDAR based approaches Huang et al. (2009); Kammel and Pitzer (2008) provide a clear "signature" (i.e. reflectiveness) of the lanes invariant to light changes. This kind of sensor, however, is rarely adopted because of its high integration cost. The data processing technology from LIDAR scanners is more complex than that of camera images. The short-range view of LIDAR also limits the ability on marking detection McCall and Trivedi (2006). Other Scanning Laser Radar (SLR) based approaches Shimomura et al. (2002); Grimmer (1993) performs similar advantages and disadvantages as LIDAR.

Vision-based sensors. Fisheye cameras Li and Shimomura (2008) are well suited for multi-marking detection thanks to wide capture range. However, the marking shapes are deformed because of distortion, making lane fitting a complex task. Stereo vision systems Pradeep et al. (2008) grant access to dense 3D data with the use of two cameras where road structure, such as road curbs, can be detected. Such vision systems are regarded as a compromise between monocular camera and LIDAR Hillel et al. (2014). The detecting range of stereo vision is deeper than LIDAR, while the range accuracy and reliability abilities of stereo vision are weaker.

As the most widely used perception modality, monocular vision remains a mature technology at a lower cost. The information from a camera vision is closer to a driver's perception than other apparatuses. Since the road markers are designed for human vision, it is appropriate to detect lane markings using monocular vision. The sensitivity of vision systems to light changes can be alleviated by image processing algorithms as stated in Aly (2008). In this study, a facing-forward camera on board of a vehicle is

adopted.

Other modalities. Geographic Information System (GIS) and GPS provide location and environment information to assist marking detection. In Wang et al. (2012), a robust hybrid approach to Predictive Lane Detection (PLD) is proposed, which refers to vehicle localization on offline digital maps to estimate road geometry, which offers strong cues to limit the lane marking model parameters. In Schule et al. (2013), a sensor fusion system combines digital map information and camera sensors to estimate the 3D road course. GIS and GPS are considered as supplement of traditional lane detection, the road curve tendency can be predicted if the precision of digital map and localization is adequate. However, the low cost GPS receivers achieve an accuracy of 5-10 meters Wing et al. (2005), the precision of open source map data reach up to 5 meters Ather (2009); Haklay (2010). Hence, the development of map based marking localization strongly depends on the improvement of digital map and GPS accuracy. In Seo and Rajkumar (2014); Son et al. (2013), vehicle dynamics (i.e. vehicle direction) are elaborated to help marking detection. In Janda et al. (2013); Huang et al. (2009), vision and Radar based perception modalities are fused together to improve feature extraction efficiency.

Vehicle dynamics also help lane detection. Vehicle internal sensors measure vehicle yaw rate, vehicle speed and acceleration, which are often used in temporal integration module to enable marking model tracking McCall and Trivedi (2006); Labayrade et al. (2006). The detection of these approaches lies on the measurement accuracy, and IMU can replace the internal sensors Huang et al. (2009).

2.1.3 Preprocessing

The main task of preprocessing step is to prepare the input images for image processing: alleviating or eliminating noisy effects, pruning image parts and recovering image information.

Noisy effects reduction. The most common ways to reduce noise is to binarize the image with a predefined threshold He et al. (2004); Soquet et al. (2007). But fixed thresholds have nothing to do to cope with variable illumination in road conditions (e.g. weather changing, shadows), publications such as Lu et al. (2008); Soquet et al. (2007) use adaptive threshold to deal with this problem. Blurring the image Truong and Lee (2008); Borkar et al. (2009a) can be used to reduce random and pepper-and-salt noises.

To reduce image load and improve image resolution, color channel conversion is employed. Works such as Cheng et al. (2006); Katramados et al. (2009); Álvarez et al. (2007) perform a variety of color-space transformations to Hue, Saturation, Lightness (HSL), YCbCr and others. In Li et al. (2004), R and B channels are used to form gray-level image. Besides, Red, Green, Blue (RGB) to Hue, Saturation, Intensity (HIS)

Rotaru et al. (2004); Samadzadegan et al. (2006) and RGB to Hue, Saturation, Value (HSV) conversions are discussed.

Finlayson-Hordley-Drew (FHD) algorithm Assidiq et al. (2008) and high-pass filters Ishikawa et al. (2003) are used to cast strong shadows. In McCall and Trivedi (2006), road edges are extracted to filter shadow-related clutters.

Image parts pruning. To remove part of an image is also part of preprocessing step. Some works Cheng et al. (2006); Yamaguchi et al. (2008) focus on detecting road obstacles and omitting the area of obstacles in the image. In Huang et al. (2009); Hernández et al. (2009), 3D data from LIDAR is carried out to reject obstacles.

Considering object detection can be unreliable because of high False Positive rate, ROI is an alternative to reduce image size. Several ways are used to split ROI: using a predefined percent of image from bottom to top Hu et al. (2004); Yu et al. (2008), introducing vanishing point and vanishing line Sun et al. (2006); Wen et al. (2008), dividing the image horizontally Bellino et al. (2005) or vertically Zhang et al. (2013); Kang et al. (1996).

IPM technology Nieto et al. (2007); Tuohy et al. (2010) has been utilized in lane detection for two reasons. On one hand, a pair of lane markings is nearly-parallel in BEV space, which brings computational convenience. On the other hand, the backgrounds (i.e. sky, roadside) are not considered in IPM.

Image recovery. Some road features may be blocked or faded, but appear in a series of continuous frames. In Wu et al. (2008); Ogawa and Takagi (2006), these missing features are estimated using image recovery methods. Temporal blurring can be also used to retrieve missing or discontinuous road markings Borkar et al. (2009a).

2.1.4 Feature Types

Numerous types of features are contained in a road environment image. In feature extraction step, lane marking features are retrieved in two steps: feature selection, and feature extraction. The former is to determine a suitable feature, which is prone to detect, while the latter implements methods to discover these features pixels. The common used features in lane marking detection applications are color, color gradient, shape, and texture.

Color. Normally, lane markings are painted with remarkable colors (i. e. white, yellow, orange) to mark lane areas. So color is the most common property for marking detection. Even in gray scale images, the markings tend to perform higher gray value than road surfaces. However, color based descriptors do not deal with light changes and is sensible to noises. Color features are captured by color-based extraction methods mentioned in Section 2.1.5.

Color gradient. Color gradient is regarded as the color or gray scale differences be-

tween neighboring pixels. On road scenes, the marking color and road surface color vary with the change of nature light intensity (i.e. a building shadow, a weather change). It is difficult to determine a fixed threshold of color in outdoor environments. However, the color gradient almost stays steadily because no matter how the illumination condition varies, the marking color and surface color value float simultaneously. One defect of color gradient is feature misleading caused by objects. For instance, vehicles on road can be detected as False Positive lane marking features because of the color difference between vehicle bodies and road surfaces. Histogram of Oriented Gradients (HOG) descriptor Dalal and Triggs (2005), a gradient based method, is implemented to retrieve lane markingsLinarth and Angelopoulou (2011).

Shape. The color based features own their own disadvantages, so marking shape can be considered as an alternative feature. In perspective space, the marking shape on an image is trapezoid-like, which can be extracted by a shape template. In BEV space, a pair of lane markings is quasi-parallel, bringing convenience in marking detecting.

Texture. Texture plays an important role in human visual perception and provides vital information for recognition and interpretation Chindaro et al. (2003). This feature is seldom exploited by monocular vision, because the difference of marking surface and road surface is hard to distinguish through single camera. Texture is used to retrieve road surface and lane marking using 3D perception sensors, such as stereoscopic Danescu and Nedevschi (2009); Pradeep et al. (2008) or LIDAR Huang et al. (2009); Kammel and Pitzer (2008).

Multi-feature. The usage of single type of feature confronts limitation mentioned above. The combination of two or more features is possible to increase the accuracy of feature extraction. In Li and Nashashibi (2011), *a priori* knowledge of lane marking segment is proposed, which includes features of a whole marking segment, not only marking pixels. In Shi et al. (2009); Lipski et al. (2008), the brightness change and marking shape are considered together.

2.1.5 Extraction Methods

Kinds of extraction methods are classified according to different feature types in this part.

Color extraction. In Cheng et al. (2006), authors employ three multivariate Gaussian distributions for white, yellow, and red lane markings to represent the three main classes of lane-marking colors. Facing the illumination variation, a threshold of gray scale or color for an image is predicted according to LIDAR Ogawa and Takagi (2006), history images Borkar et al. (2009b) or offline procedure Cheng et al. (2006); Kornhauser (2007). To further resist the impact from nature lights, a recursive Bayesian segmentation algorithm is demonstrated in Nieto et al. (2011). The color range of

pavement, markings, and other objects are determined with respect to the Gaussian distribution on histograms of image pixels.

Gradient extraction. Extraction methods of color gradient work with the prerequisite of obvious difference in appearance between markings and road surfaces. Simple gradients are computed in Sawano and Okada (2006); Nieto et al. (2008). To enable the directional response at specific directions, steerable filter is introduced in McCall and Trivedi (2006) and improved in Liu and Li (2013). Other frequently used gradient based methods are Canny filter Wang et al. (2004); Truong et al. (2008), Sobel filter Lai and Yung (2000); Yim and Oh (2003), peak finding in scan-line Huang et al. (2004); Wang et al. (2005) and thresholding Benmansour et al. (2008); Nieto et al. (2008).

Shape extraction. To discover low-high-low intensity pattern along image rows is a common strategy Huang et al. (2009); Labayrade et al. (2006); Wu et al. (2008) of shape based extraction methods. Labayrade et al. (2006) convolved the image with a step filter and then searched for couples of adjacent responses with opposite signs. Specifically in perspective space, the kernel of shape filter changes in different rows Huang et al. (2009); Kornhauser (2007). A more commonly practiced technique McCall and Trivedi (2006); Jiang et al. (2009) circumvents the need for varying kernels by first warping the image in a manner that compensates the perspective effect. In BEV space, the road markings are approximated as quasi-vertical white lines surrounded by black environment. Methods in Bertozzi and Broggi (1998); Sehestedt et al. (2007) are implemented to treat with this problem. Ridge properties of a single marking is extracted in Kang et al. (2014); Hur et al. (2013)

Texture extraction. Most texture extraction methods Huang et al. (2009); Danescu and Nedevschi (2009) aim to road detection rather than marking detection. LIDAR is responsible to find different surfaces with different reflectiveness, which are meaningful to search for markings Labayrade et al. (2006); Nieto et al. (2008). In Seibert et al. (2013), the road shoulders, curbs and guardrails are localized via a texture based area classification.

Multi-feature extraction. Other approaches Enzweiler et al. (2013) rely on multi-cue fusion for increasing robustness in advance. For example, in Apostoloff and Zelinsky (2003), color, color gradient and road shape are combined to model a confidence in a particle filter. In Jiang et al. (2011), marking features are derived according to a shape based method, and outliers are eliminated through the major direction of the blobs in an image.

2.1.6 Marking Models

The main goal of this stage is to estimate a high-level representation of the road marking, which can be used by the succedent work.

Parametric model.A parametric model is one in which the indexing parameter is a finite-dimensional vector, for instance, straight lines Borkar et al. (2009b); Kong et al. (2009), parabolic curves McCall and Trivedi (2006); Huang et al. (2009), circumference arcs Nieto et al. (2011); Jiang et al. (2010), hyperbola Wang et al. (2008); Zhu et al. (2008).

Non-parametric model. Non-parametric models differ from parametric models in that the model structure is not specified a priori but is instead determined from data. i. e. continuous pixels, but not necessarily smooth Broggi and Cattani (2006); Nefian and Bradski (2006).

Semi-parametric model. A semi-parametric model is a model that has parametric and nonparametric components. Popular models of this type include linear parabolic model Liu et al. (2011), piecewise linear model or poly-line model Huang et al. (2009); Sawano and Okada (2006); Kim (2008), hyperbolic polynomial Labayrade et al. (2006) and clothoid Danescu et al. (2007); Nedevschi et al. (2006). Specifically, spline models contain various appearances, such as Bezier spline Aly (2008), B-spline Wang et al. (2004) , cubic spline Kim (2008) and cubic hermit spline Huang et al. (2009).

Comparing to parametric models, semi-parametric models have the advantage of a free shape, which means that the model shape is not a specific geometry. On the other side, semi-parametric model fitting may lead to over-fitting and unrealistic marking curvatures. Poly models can also bring more parameters to fit, which increase processing time.

In perspective space, straight line is the most popular model Li et al. (2013). However, the marking shapes become complex in this vision, which increases fitting complexity. BEV space has the advantage to present the actual shapes of an object from a top view. Generic circumference Nieto et al. (2011) and parabola Lu et al. (2014a) have better performances to simulate lane markings. To make full use of IPM view, vehicle dynamics (i.e. vehicle direction, vehicle position) are added into lane marking model in Sivaraman and Trivedi (2013).

The core of marking model selection is the tradeoff between model complexity (amount of model parameters) and computational calculation cost. A simple model owns fewer model parameters, and therefore brings convenience in model fitting, but a simple model is not able to simulate the road marking shape, and tends to be affected by noise features; On the contrary, a complex model have more parameters to fit various marking shapes, but the model estimation run time raises with the increase of parameter amount. Typical on-vehicle cameras provide a significant perspective effect at nearer distance and poor vision at a farther distance, a complex shape model results in no additional accuracy in marking detection area. Empirically, a three parameter shape model (circumference, parabola) is enough for road marking fitting Nieto et al. (2012).

2.1.7 Model Regression

Robust estimation techniques are employed to fit the parametric and semi-parametric models. The aim is to retrieve a precise marking shape among a number of feature candidates. Meanwhile, the estimation method is robust to noise pixels. The algorithms of model regression used in marking detection are classified into three categories.

Parameters estimation methods. The most classical way to handle model fitting problem is Least Square Method (LSM). LSM is exploited to estimate parabolic curves in McCall and Trivedi (2006) and circumference arcs in Jiang et al. (2010). Weighted LSM Tarel et al. (2002), which is an improved LSM algorithm, is introduced to marking detection in Labayrade et al. (2006).

Robust iterative methods. LSM is not robust because outliers may become dominating with respect to the actual model. The robust iterative methods modify the objective function so that the influence of residues is limited. The principal consequence is a slower convergence speed Malis and Marchand (2006). For instance, M-estimator Huber (2011) is to modify the objective function of LSM by penalizing the largest residues. In Bai et al. (2008), M-estimator is used for detecting marking.

Robust voting methods. Voting methods are well adapted to problems with a significant number of feature pixels compared to the amount of model parameters. Hough transform Hough (1959) is a frequently used robust voting method in marking detection Romdhane et al. (2011); Javadi et al. (2012). Some modified methods are proposed in Illingworth and Kittler (1988). Hough transform based methods are applied in lane marking detection areas in Javadi et al. (2012); Liu et al. (2011).

The parameter amount for Hough transform based methods to estimate should be within 4 parameters, otherwise the computation time become rapidly prohibitive. RANdom SAmple Consensus (RANSAC) Fischler and Bolles (1981) is a probabilistic voting method to reduce the calculation time cost of classical voting approaches such as Hough transform. The disadvantage of RANSAC is that improper tuned parameters leads to wrong fitting results. RANSAC is used to fit parametric models and semi-parametric models in Kang et al. (2014); Borkar et al. (2009b).

2.2 Image Processing

In image processing part, the camera image is mapped into BEV space at first, a second-order derivative filter is utilized to extract marking features, and a cell-based blob algorithm is employed to eliminate potential outlier features.

2.2.1 Inverse Perspective Mapping (IPM)

The road images are captured by a monocular camera, which is modeled using a pinhole projective model assuming no distortion and zero skew. The images are processed

through Inverse Perspective Mapping (IPM). This transformation provides vertical and paralleled lane markings in BEV space, which greatly facilitates marking detection strategies. To derive the road image in BEV space, bi-linear interpolation is introduced. The idea is to calculate gray scale of a pixel in BEV space according to corresponding pixels in perspective space. The mapping between perspective space and BEV spaces is detailed in Appendix A.

The corresponding rational point (x_{per}, y_{per}) derived from Eq. A.11 according to integer points (x_{bev}, z_{bev}) can be expressed as $(i_{per}+u_{per}, j_{per}+v_{per})$, where i_{per}, j_{per} are integers, and u_{per}, v_{per} are decimals. (x_{per}, y_{per}) is not an existing pixel in the camera vision. So the gray scale value of (x_{bev}, z_{bev}) is determined by the four pixels around (x_{per}, y_{per}), as depicted in Fig. 2.3. The bi-linear interpolation is indicated as:

$$\begin{aligned} g_p(x_{bev}, y_{bev}) =& (1-u_{per}) \cdot (1-v_{per}) \cdot g_p(i_{per}, j_{per}) + (1-u_{per}) \cdot v_{per} \cdot g_p(i_{per}, j_{per}+1) \\ &+ u_{per} \cdot (1-v_{per}) \cdot g_p(i_{per}+1, j_{per}) + u_{per} \cdot v_{per} \cdot g_p(i_{per}+1, j_{per}+1), \end{aligned} \tag{2.1}$$

where $g_p(x, y)$ is the gray scale of pixel (x, y) in perspective space.

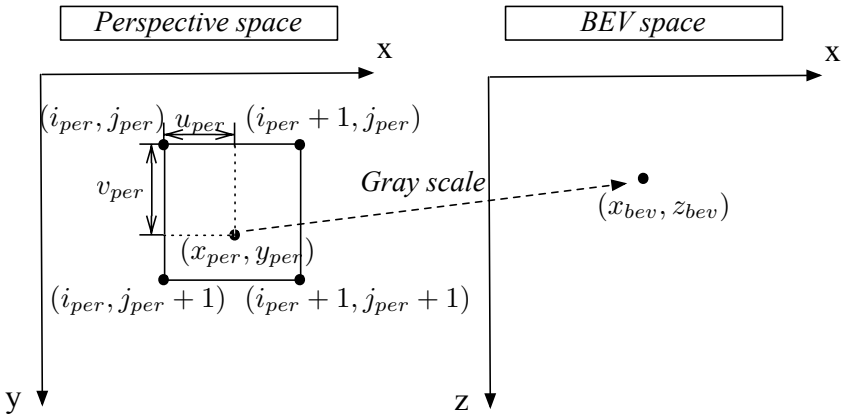


Figure 2.3: *Bi-linear interpolation in perspective space. The gray scale value of a pixel in BEV space counts on four corresponding pixels in perspective space.*

Therefore, the perspective images are transmitted to BEV images, which are used for further image processing steps.

2.2.2 Second-order Derivative Filter

A second order derivative filter along the horizontal direction is applied to process BEV images where lane markings are nearly vertical. This filter is presented as follows:

$$I_{xx} = G_x * G_x * I_{ipm}, \tag{2.2}$$

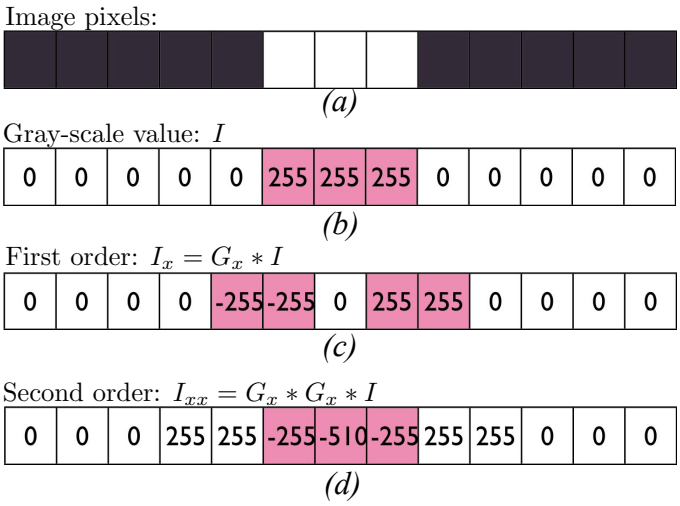


Figure 2.4: *Second-order derivative filter. In image pixels line, white blocks are lane marking pixels, and black blocks are road surface. In the other three lines, pink blocks represent distinct pixels.*

where I_{ipm} is input image after IPM transformation, $G_x = [+1, 0, -1]$ is horizontal Sobel descriptor, I_{xx} is the filtered image.

Second-order derivative is able to detect "black-white-black" transition. Fig. 2.4 depicts an example of this filter. Fig. 2.4a is a row of the BEV image along x-axis with lane marking pixels and road surface pixels. A white square represents lane marking pixels, and a gray square is road surface. To explain the principle of second-order derivative filter, we assume that the 8-bit gray scale value of marking pixels is 255 (white), and the gray scale value of road surface is 0 (black), as shown in Fig. 2.4b. When operator G_x convolves on input image I_{ipm}, pixels with high value cannot represent the lane markings, as illustrated in Fig. 2.4c. When operator G_x convolves again on I_{ipm}, the result is as Fig. 2.4d. "Black" pixels near "white" pixels have high positive value, and "white" pixels own high negative value. So the lane marking pixels can be determined by the convolving value. Second-order derivative filter can detect "white" pixels surrounded by "black" pixels, if the width of the "white" part is less than 5 pixels. Assuming that a marking width is between 10-35 cm in reality, the width of projected marking in BEV image is 1-4 pixel, with the resolution 0.1m/pixel. So second-order derivative filter is able to detect the marking pixels in BEV images. In addition, time consuming of this method is much less than complex extraction methods such as CANNY.

2.2.3 Cell-based Blob Algorithm

To eliminate the outliers included in the image with respect to the information from the whole figure, a cell-based blob algorithm inspired from Jiang et al. (2011), is introduced. A blob is a group of connected "white" pixels. The BEV image is divided into 4×4 cells

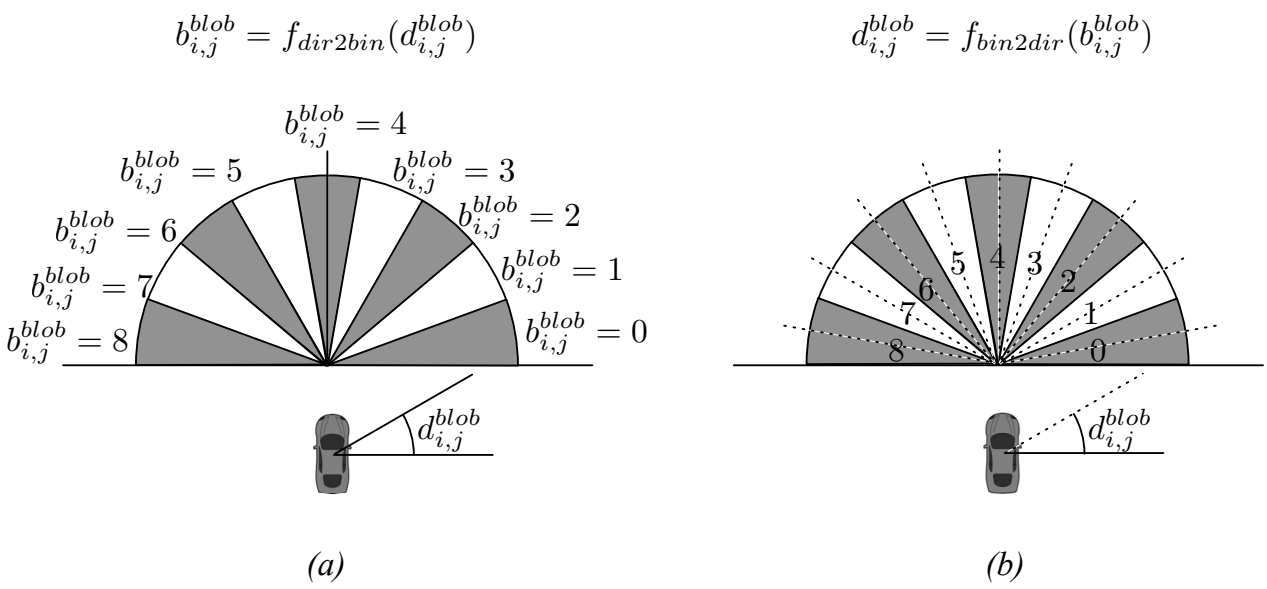


Figure 2.5: *Bin level definition according to $d_{i,j}^{blob}$. (a) Determine $b_{i,j}^{blob}$ with respect to $d_{i,j}^{blob}$; (b) determine $d_{i,j}^{blob}$ with respect to $b_{i,j}^{blob}$.*

at first, as depicted in Fig. 2.6a, then all the blobs in the image are searched. When the j^{th} blob in i^{th} cell is detected, the blob direction $d_{i,j}^{blob}$ can be computed according to the angle between the top left pixel and the bottom right pixel of the blob. The blob width $w_{i,j}^{blob}$, defined as the average width of the blob along x-axis, is computed at the same time. The blob directions are then classified into 9 bin levels (20° per bin) according to:

$$b_{i,j}^{blob} = f_{dir2bin}(d_{i,j}^{blob}), \tag{2.3}$$

which is defined as Fig. 2.5a. From the image, it implies that each $d_{i,j}^{blob}$ has its own corresponding bin level $b_{i,j}^{blob}$.

The entire bin levels of a cell can be used to vote for the main bin level of the current cell, as shown in Fig. 2.6a. When all the bin levels $b_{i,j}^{blob}$ in cell i are derived, number of blobs of each bin level is determined, the bin level of the host cell b_i^{cell} can be voted according to the maximum bin level number. When bin levels of all the cells b_i^{cell} $(i = 1, 2..., 16)$ in an image are calculated, the main bin level of the current image b^{image} is derived. The allowed direction range of the image $\left[D_{min}^{image}, D_{max}^{image}\right]$ is computed according to b^{image} as:

$$\left[D_{min}^{image}, D_{max}^{image}\right] = \left[f_{bin2dir}\left(b^{image}\right) + \triangle D_{blob}, f_{bin2dir}\left(b^{image}\right) - \triangle D_{blob}\right], \tag{2.4}$$

where $f_{bin2dir}$ is a mapping from bin level to vehicle direction defined in Fig. 2.6b, $\triangle D_{blob}$ is a direction range constant.

Meanwhile, a bin level template, as depicted in Fig. 2.6c, is constructed offline according

to various detected marking results. In this template, possible bin sequences of each cell is fixed as $\left[B^{tpl}_{i,min}, B^{tpl}_{i,max}\right], i = 1, ..., 16$. Specifically, the assignment is:

$$\left[B^{tpl}_{i,min}, B^{tpl}_{i,max}\right] = \begin{cases} [7,7] & (i=1) \\ [2,6] & (i=2,3,6,7) \\ [1,1] & (i=4) \\ [7,8] & (i=5) \\ [0,1] & (i=8) \\ [6,7] & (i=9,13) \\ [4,6] & (i=10,11,14,15) \\ [1,2] & (i=12,16) \end{cases}, \tag{2.5}$$

where $B^{tpl}_{i,min}$ and $B^{tpl}_{i,max}$ are minimum and maximum bin sequence, i is cell sequence arranged as Fig. 2.6e. This template represents possible bin levels in different cells, according to real world road environment. For instance, as in Fig. 2.6a, left-heading markings of current lane appear only at the left side part of a BEV image, and is unable to appear at the right side part.

Finally, outlier blobs are excluded according to three conditions:

$$d^{blob}_{i,j} \in \left[D^{image}_{min}, D^{image}_{max}\right], \tag{2.6}$$

$$b^{blob}_{i,j} \in \left[B^{tpl}_{i,min}, B^{tpl}_{i,max}\right], \tag{2.7}$$

$$w^{blob}_{i,j} \in \left[W^{blob}_{min}, W^{blob}_{max}\right], \tag{2.8}$$

which are the main bin level of the image, the bin level template, and the size of blob. In Eq. 2.8, W^{blob}_{min} and W^{blob}_{max} are the suggested blob width range.

The output result is shown in Fig. 2.6d. And the cell-based blob algorithm is demonstrated as Algorithm 2.2.1.

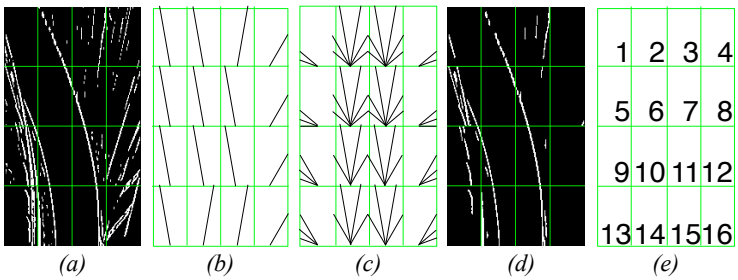


Figure 2.6: *Blob algorithm. (a) Input binary image; (b) voted bin levels in each cell; (c) bin level template; (d) output image; (e) cell sequence.*

Algorithm 2.2.1 Cell-based blob algorithm

Input: - binary BEV image I_{xx}

Output: output image I_{blob} with few outliers

1: ▶ Define vector<cell_structure> *cell_storage*
2: **for** $i = 1; i <= 16; i++$ **do**
3: ▶ Define vector<cell_structure> *current_cell*
4: **repeat**
5: ▶ $j = j + 1$
6: ▶ calculate $d_{i,j}^{blob}$, $b_{i,j}^{blob}$, $w_{i,j}^{blob}$
7: **if** $w_{i,j}^{blob} \in [W_{min}, W_{max}]$ **then** ▷ meet Eq. 2.8
8: ▶ save blob pixels, $d_{i,j}^{blob}$, $b_{i,j}^{blob}$ to *current_cell*
9: **end if**
10: **until** all the blobs are detected
11: ▶ $j = 0$
12: ▶ vote for the cell bin level b_i^{cell} and save to *current_cell*
13: ▶ save *current_cell* to *cell_storage*
14: **end for**
15: ▶ vote for the main bin level b^{image} of image I_{xx}, according to b_i^{cell} $(i = 1, 2..., 16)$ in *cell_storage*
16: ▶ get $\left[D_{min}^{image}, D_{max}^{image}\right]$ with respect to b^{image}
17: **for** $i = 1; i <= 16; i++$ **do**
18: ▶ define vector<cell_structure> *current_cell*
19: ▶ copy parameters of current cell from *cell_storage* to *current_cell*
20: **for** $j = 1; j <= blob\_number; l++$ **do**
21: **if** $d_{i,j}^{blob} \notin \left[D_{min}^{image}, D_{max}^{image}\right] || b_{i,j}^{blob} \notin \left[B_{i,min}^{tpl}, B_{i,max}^{tpl}\right]$ **then** ▷ meet Eq. 2.6 and Eq. 2.7
22: ▶ remove j^{th} blob in *current_cell*
23: **end if**
24: **end for**
25: ▶ save *current_cell* to *cell_storage*
26: **end for**
27: ▶ process *cell_storage* to provide I_{blob}

At the end of image processing part, a binary image is produced. White pixels represent lane marking candidates. Considering the complexity and diversity of outdoor traffics, some noise pixels still exist in the image. The next section explains approaches to estimate lane marking parameters from an $M_{bev} \times N_{bev}$ binary image with a few noise pixels.

2.3 Lane Marking Model Estimation

A parabola: $x = c + d \cdot y + e \cdot y^2$ is chosen as marking model in BEV space. On one side, the marking shape variation is restricted because of IPM range in our method, so a parabola is more suitable than complex models (i.e. spline-like model). On the other side, a straight line model, which appears frequently in real situations, is included in parabola model. The marking initialization step determines the zero order component c. The first and second order components d and e are estimated through an improved multi-kernel estimation (MKE) method with hierarchical weights.

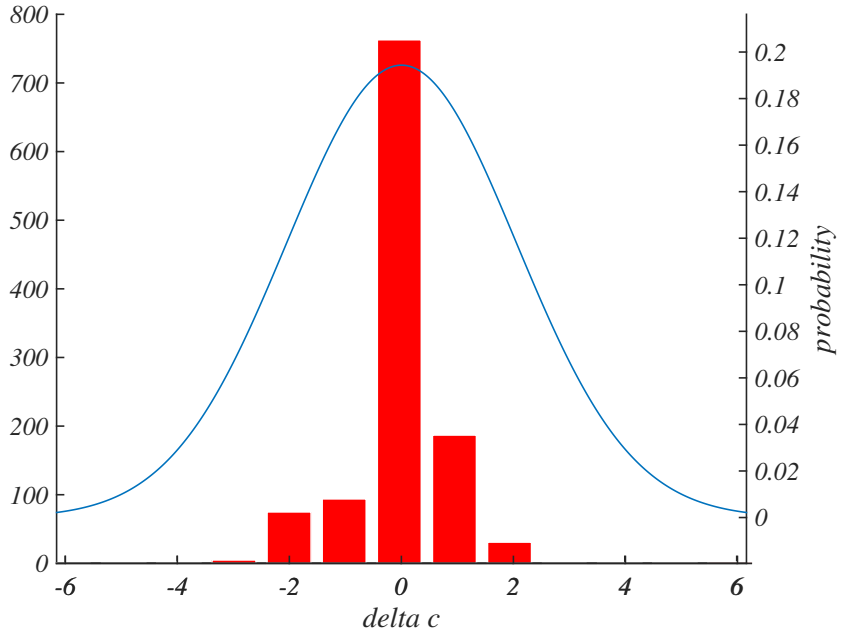


Figure 2.7: *Gaussian model fitting on* Δc.

2.3.1 Initialization

This part aims to determining the zero order components c_l and c_r of left and right markings together using a parametric Gaussian model based method. To this end, the intersections of both lane markings at x-axis in BEV images are estimated using multi-cues. These two intersections are exactly model parameters c_l and c_r.

At first, several parameters used in the initialization method are estimated offline, using massive detection results. One estimation is to fit a Gaussian model on the distribution of zero order component c variation. The variation Δc is defined as:

$$\Delta c = c\,(t) - c\,(t-1)\,, \tag{2.9}$$

The histogram of Δc is illustrated in Fig. 2.7, red columns are experimental data, while

the blue curve is the fitted Gaussian curve, whose normalized expression is:

$$X_{\Delta c}^{ini} \sim \mathcal{CN}\left(0, \left(\sigma_1^{ini}\right)^2\right). \tag{2.10}$$

Another estimation is the empirical distributions of c_l and c_r, as in Fig. 2.9.

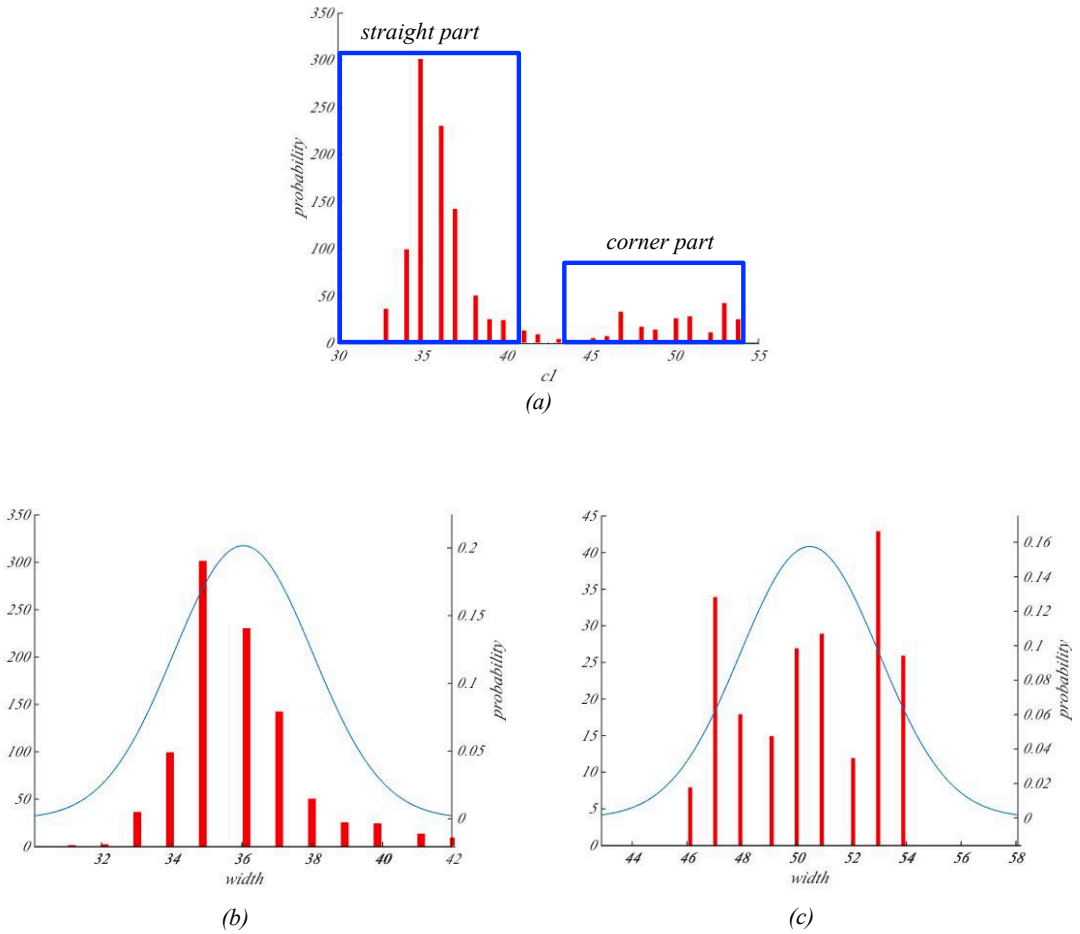


Figure 2.8: *Distribution of road lane width. (a) Lane width distribution; (b) straight lane part; (c) corner lane part.*

Gaussian models are fitted, the normalized equations are:

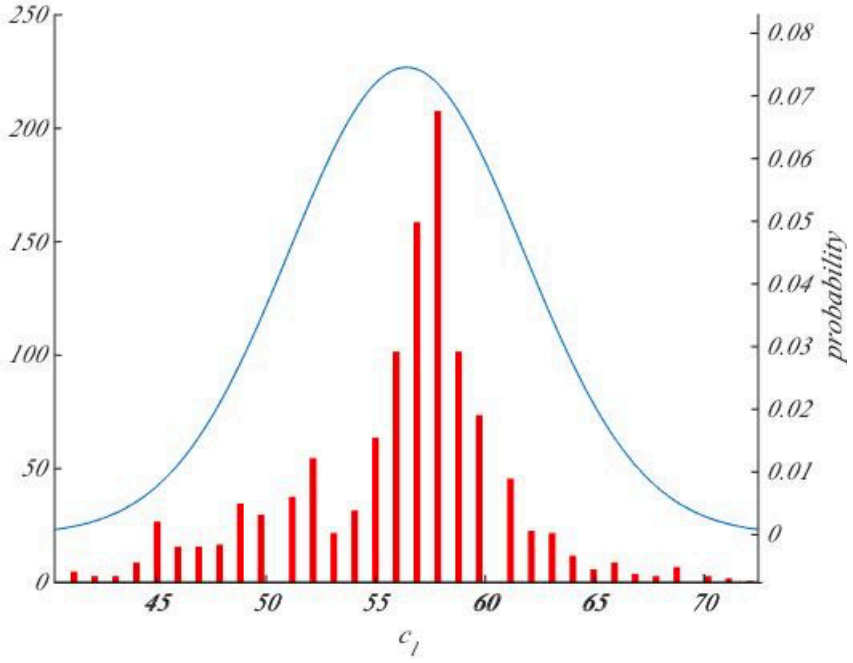
$$X_{c_l}^{ini} \sim \mathcal{CN}\left(\mu_{4,l}^{ini}, \left(\sigma_{4,l}^{ini}\right)^2\right), \tag{2.11}$$

$$X_{c_r}^{ini} \sim \mathcal{CN}\left(\mu_{4,r}^{ini}, \left(\sigma_{4,r}^{ini}\right)^2\right). \tag{2.12}$$

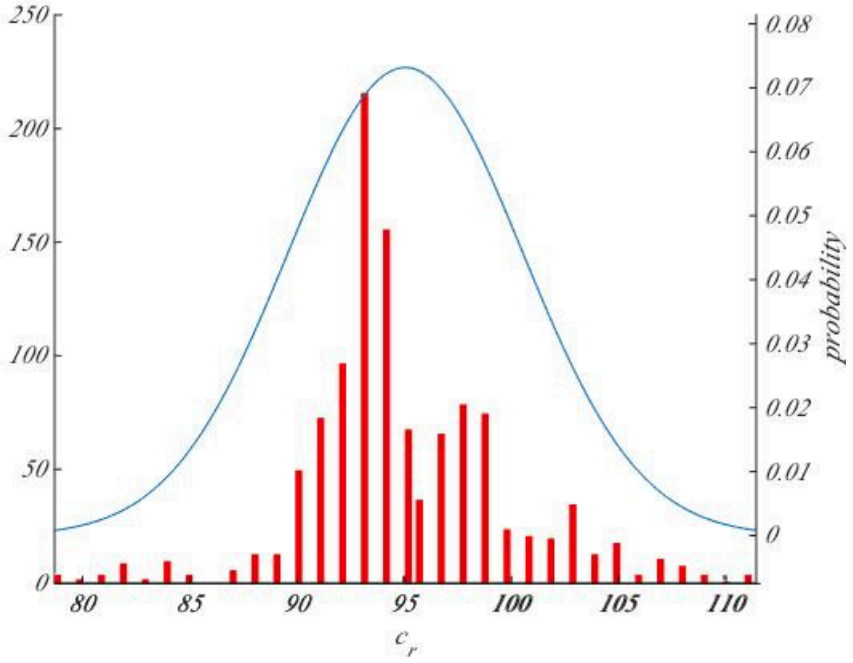
The last offline estimation is the distribution of road lane width, as Fig. 2.8. The lane width is defined as:

$$d_{lane} = |c_r - c_l|. \tag{2.13}$$

When the lane width distribution is extracted as Fig. 2.8a, the histogram is divided



(a)



(b)

Figure 2.9: *Empirical distribution of c_l and c_r. (a) Empirical distribution of c_l; (b) Empirical distribution of c_r.*

into two parts, width of straight lane and width of corner lane. The Gaussian model estimation is implemented in Fig. 2.8b and Fig. 2.8c for these two types separately. The normalized functions are:

$$X_{str}^{ini} \sim \mathcal{CN}\left(d_{str}, \left(\sigma_{5,str}^{ini}\right)^2\right), \tag{2.14}$$

$$X_{cor}^{ini} \sim \mathcal{CN}\left(d_{cor}, \left(\sigma_{5,cor}^{ini}\right)^2\right). \tag{2.15}$$

After offline model estimation, several parameters used in initialization step are derived. It is possible to estimate the zero order components of lane marking models. A Region Of Interest (ROI) I_{ROI} with the size of $M_{ROI} \times N_{bev}$, is cropped from the binary image, as shown in Fig. 2.10a. The length M_{ROI} should be set carefully. The markings cannot be approximated as straight lines if M_{ROI} is too long, the discontinuous markings do not appear in I_{ROI} if M_{ROI} is too short. Then the following distributions are considered: previous detection distribution X_1^{ini}, white pixels distribution X_2^{ini}, Hough lines distribution X_3^{ini}, prior data distribution X_4^{ini} and lane width distribution X_5^{ini}. The normalized distributions $\overline{X_1^{ini}}$ to $\overline{X_5^{ini}}$ are shown as in Fig. 2.10c and Fig. 2.10d.

Normally, in BEV space the intersections move smoothly along the first row of I_{ROI} if the input frame frequency is high enough. So the positions of both intersections are strongly related to the previous position c_l^{pre} and c_r^{pre}. Distribution X_1^{ini} is therefore represented as:

$$X_1^{ini} \sim 1/2 \cdot \left[\mathcal{CN}\left(c_l^{pre}, \left(\sigma_1^{ini}\right)^2\right) + \mathcal{CN}\left(c_r^{pre}, \left(\sigma_1^{ini}\right)^2\right)\right], \tag{2.16}$$

where σ_1^{ini} is computed in Eq. 2.10.

White pixels at the bottom row of I_{ROI} are also potential candidates. The white pixels distribution is:

$$X_2^{ini} \sim 1/n_w^{ini} \cdot \left[\mathcal{CN}\left(\mu_{w,1}^{ini}, \left(\sigma_2^{ini}\right)^2\right) + \cdots + \mathcal{CN}\left(\mu_{w,n_w^{ini}}^{ini}, \left(\sigma_2^{ini}\right)^2\right)\right], \tag{2.17}$$

where $\mu_{w,1}^{ini}, \cdots, \mu_{w,n_w^{ini}}^{ini}$ are locations of the white pixels along x-axis.

Considering that dashed markings may have no white pixel cues, for instance the left marking in Fig. 2.10a, Hough transform is helpful to find the intersections of these discontinuous markings. Let $\mu_{ht,1}^{ini}, \cdots, \mu_{ht,n_{ht}^{ini}}^{ini}$ denote the intercepts of these Hough lines on x-axis, the Hough line distribution is:

$$X_3^{ini} \sim 1/n_{ht}^{ini} \cdot \left[\mathcal{CN}\left(\mu_{ht,1}^{ini}, \left(\sigma_3^{ini}\right)^2\right) + \cdots + \mathcal{CN}\left(\mu_{ht,n_{ht}^{ini}}^{ini}, \left(\sigma_3^{ini}\right)^2\right)\right]. \tag{2.18}$$

The prior distribution $\mathcal{CN}\left(\mu_{4,l}^{ini}, \left(\sigma_{4,l}^{ini}\right)^2\right)$ and $\mathcal{CN}\left(\mu_{4,r}^{ini}, \left(\sigma_{4,r}^{ini}\right)^2\right)$ are fitted offline ac-

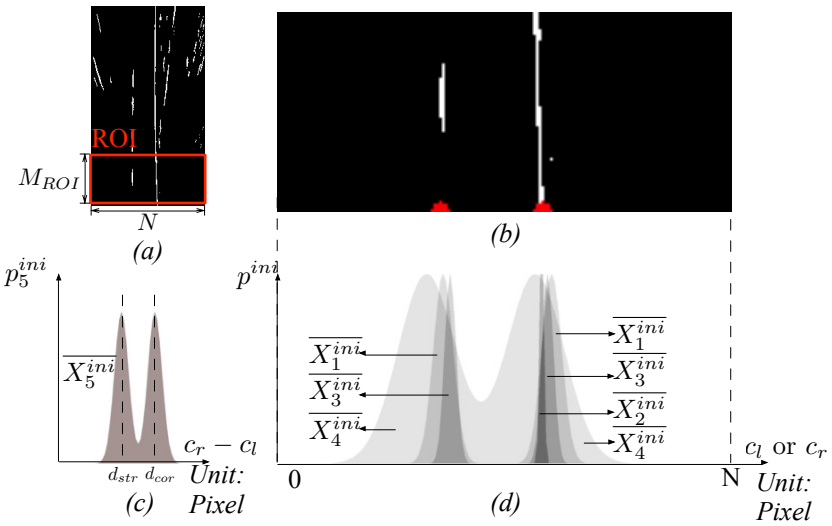


Figure 2.10: *Parametric Gaussian model based Initialization. (a) ROI, (b) amplified ROI, (c) lane width distribution $\overline{X_5^{ini}}$, (d) Gaussian distributions $\overline{X_1^{ini}}$ to $\overline{X_4^{ini}}$ corresponding to (b). Estimated results c_l^* and c_r^* are depicted as two red dot in (b).*

cording to numerous historical data which is defined as:

$$X_4^{ini} \sim 1/2 \cdot \left[\mathcal{CN}\left(\mu_{4,l}^{ini}, \left(\sigma_{4,l}^{ini}\right)^2\right) + \mathcal{CN}\left(\mu_{4,r}^{ini}, \left(\sigma_{4,r}^{ini}\right)^2\right)\right], \tag{2.19}$$

where $\mu_{4,l}^{ini}$, $\sigma_{4,l}^{ini}$, $\mu_{4,r}^{ini}$ and $\sigma_{4,r}^{ini}$ are estimated in Eq. 2.11 and Eq. 2.12.

The lane width distribution is represented as:

$$X_5^{ini} \sim 1/2 \cdot \left[\mathcal{CN}\left(d_{str}, \left(\sigma_{5,\mathrm{str}}^{ini}\right)^2\right) + \mathcal{CN}\left(d_{cor}, \left(\sigma_{5,\mathrm{cor}}^{ini}\right)^2\right)\right], \tag{2.20}$$

where d_{str} and d_{cor}, computed in Eq. 2.14 and Eq. 2.15, denote the average lane width of straight sections and corner sections respectively. The probability distribution of c_l and c_r is then given as:

$$p_t^{ini}(c_l, c_r) = \left[\sum_{j=l,r}\left(\sum_{i=1}^{3} k_i^{ini} \cdot p_i^{ini}(c_j)\right) \cdot p_4^{ini}(c_j)\right] \cdot p_5^{ini}(c_l - c_r), \tag{2.21}$$

where $p_i^{ini}(x)$ is probability density of $\overline{X_i^{ini}}$, k_i^{ini} is a coefficient of $p_i^{ini}(x)$. The probability distribution $p_t^{ini}(c_l, c_r)$ of the input image Fig. 2.10a is shown in Fig. 2.11.

Finally the intersections of left and right lane markings c_l^* and c_r^*, are determined as follows:

$$c_l^*, c_r^* = \arg\max_{c_l, c_r} p_t^{ini}(c_l, c_r). \tag{2.22}$$

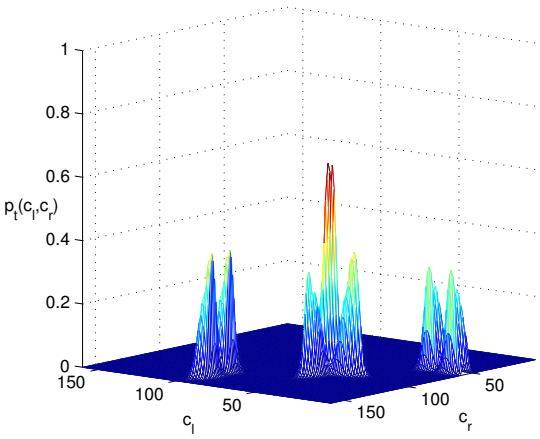


Figure 2.11: *The probability distribution of* $p_t^{ini}(c_l, c_r)$.

2.3.2 Multi-Kernel Estimation (MKE)

A multi-kernel density based method with hierarchical weights is introduced to estimate the model parameters. The process is called Statistical Hough Transform (SHT) Dahyot (2009). The parameters are modeled as multi-kernel density, and the candidates are found by comparing the probability of different line parameters. The basic descriptor of this algorithm is the similarity between an image pixel (x_i^{bev}, y_i^{bev}) and the model (c, d, e), presented as:

$$G_{pi}\left(c, d, e, x_i^{bev}, y_i^{bev}\right) = \int_{-\infty}^{+\infty} K_x^{ori}\left(x_i^{bev}\right) K_y^{ori}\left(y_i^{bev}\right) dy, \tag{2.23}$$

where,

$$K_y^{ori} = \frac{1}{\sqrt{2\pi\left(\sigma_y^{ori}\right)^2}} exp\left(-\frac{\left(y - y_i^{bev}\right)^2}{2\left(\sigma_y^{ori}\right)^2}\right), \tag{2.24}$$

$$K_x^{ori} = \frac{1}{\sqrt{2\pi\left(\sigma_x^{ori}\right)^2}} exp\left(-\frac{\left(c + dy + ey^2 - x_i^{bev}\right)^2}{2\left(\sigma_x^{ori}\right)^2}\right). \tag{2.25}$$

Gauss-Hermite quadrature method Liu et al. (2009) is employed to compute the numerical solution of G_{pi}. When G_{pi} is derived, the probability of a specified model $p_{mke}(d, e)$ is defined as:

$$p_{mke}(d, e) = \frac{1}{n_t^{mke}} \sum_{i=1}^{n_t^{mke}} w_{mke}\left(x_i^{bev}, y_i^{bev}\right) \cdot G_{pi}\left(c^*, d, e, x_i^{bev}, y_i^{bev}\right), \tag{2.26}$$

where n_t^{mke} is the total number of white pixels in the image. $w_{mke}\left(x_i^{bev}, y_i^{bev}\right)$ is a hierarchical weight, which offers corresponding coefficient according to different areas

of an BEV image. The assignment of $w_{mke}\left(x_i^{bev}, y_i^{bev}\right)$ is depicted in Fig. 2.12. In the figure, the current lane markings are more likely to appear in darker areas than in lighter areas, therefore, the pixels in darker areas are valued with a higher weight. Relatively the effects of noise pixels (e.g. vehicles on the road) in light areas are alleviated, with a lower weight.

The weight $w_{mke}\left(x_i^{bev}, y_i^{bev}\right)$ can be expresses in equation as:

$$w(x_i, y_i) = \begin{cases} 1 & 0 \leqslant y_i < \frac{3}{4}M, \frac{1}{4}N \leqslant x_i < \frac{3}{4}N \\ \frac{1}{4} & \frac{1}{2}M \leqslant y_i < M, (0 \leqslant x_i < \frac{1}{4}N \quad or \quad \frac{3}{4}N \leqslant x_i < N) \\ \frac{1}{2} & else \end{cases}. \tag{2.27}$$

The original method is modified to represent better real-time performance. At first, the kernels K_y^{ori} and K_x^{ori} in Eq. 2.24 and Eq. 2.25 are simplified using triangle model, represented as:

$$K_x^{tri} = \begin{cases} 0, & x \in (-\infty, -2\sigma_x^{ori} + x_i] \\ \frac{1}{2\sigma_x^{ori}\sqrt{2\pi\sigma_x^{ori}}} \cdot (x - x_i) + \frac{1}{\sqrt{2\pi\sigma_x^{ori}}}, & x \in (-2\sigma_x^{ori} + x_i, x_i] \\ -\frac{1}{2\sigma_x^{ori}\sqrt{2\pi\sigma_x^{ori}}} \cdot (x - x_i) + \frac{1}{\sqrt{2\pi\sigma_x^{ori}}}, & x \in (x_i, 2\sigma_x^{ori} + x_i] \\ 0, & x \in (2\sigma_x^{ori} + x_i, +\infty] \end{cases}, \tag{2.28}$$

$$K_y^{tri} = \begin{cases} 0, & y \in (-\infty, -2\sigma_y^{ori} + y_i] \\ \frac{1}{2\sigma_y^{ori}\sqrt{2\pi\sigma_y^{ori}}} \cdot (y - y_i) + \frac{1}{\sqrt{2\pi\sigma_y^{ori}}}, & y \in (-2\sigma_y^{ori} + y_i, y_i] \\ -\frac{1}{2\sigma_y^{ori}\sqrt{2\pi\sigma_y^{ori}}} \cdot (y - y_i) + \frac{1}{\sqrt{2\pi\sigma_y^{ori}}}, & y \in (y_i, 2\sigma_y^{ori} + y_i] \\ 0, & y \in (2\sigma_y^{ori} + y_i, +\infty] \end{cases}, \tag{2.29}$$

where σ_x^{ori} and σ_y^{ori} are the variance of original Gaussian kernel, $x = c + dy + ey^2$.

Then basic unit G_{pi} in Eq. 2.23 is modified from continuous function to discrete function as

$$G_{pi}(x_i^{bev}, y_i^{bev}) = \sum_{y=y_i^{bev}-2\sigma_{gpi}^{ipv}}^{y_i^{bev}+2\sigma_{gpi}^{ipv}} K_x^{tay}\left(x_i^{bev}, y_i^{bev}, y\right) K_y^{tay}\left(y_i^{bev}, y\right). \tag{2.30}$$

This modification is able to simplify the calculation in computing $G_{pi}\left(x_i^{bev}, y_i^{bev}\right)$.

The last improvement is to have a restricted range to derive $p_{mke}(d, e)$. In Eq. 2.26, the basic units $G_{pi}\left(c^*, d, e, x_i^{bev}, y_i^{bev}\right)$ of all the candidate pixels $\left(x_i^{bev}, y_i^{bev}\right)$ are summed. However, some candidate pixels provide little contributions to $p_{mke}(d, e)$. Therefore, a limitation of Eq. 2.26 is added as:

$$x_i^{bev} \in \left[c + d \cdot y_i^{bev} + e \cdot \left(y_i^{bev} \right)^2 - 3 \cdot \sigma_{gpi}^{ipv}, c + d \cdot y_i^{bev} + e \cdot \left(y_i^{bev} \right)^2 + 3 \cdot \sigma_{gpi}^{ipv} \right]. \quad (2.31)$$

Algorithm 2.3.2 Multi-kernel based algorithm

Input: - binary BEV image I_{blob}

Output: output lane marking models: d_l^*, e_l^*, d_r^*, e_r^*

1: ▶ Define parameter ranges: d_{min}^{mke}, d_{max}^{mke}, e_{min}^{mke}, e_{max}^{mke}
2: **if** lane tracking **then**
3: ▶ set d_{min}^{mke}, d_{max}^{mke}, e_{min}^{mke}, e_{max}^{mke} as local range
4: **else**
5: **if** lane detecting **then**
6: ▶ d_{min}^{mke}, d_{max}^{mke}, e_{min}^{mke}, e_{max}^{mke} set as global range
7: **end if**
8: **end if**
9: **for** $e = e_{min}^{mke}; e <= e_{max}^{mke}; e = e + e_{reso}^{mke}$ **do**
10: **for** $d = d_{min}^{mke}; d <= d_{max}^{mke}; d = d + d_{reso}^{mke}$ **do**
11: **for** $c = c_l^*$ or c_r^* **do**
12: **for** $i = 0; i < I_{blob}.row; i++$ **do**
13: **for** $j = c + d \cdot i + e \cdot i^2 - 3 \cdot \sigma_{gpi}^{ipv}; j < c + d \cdot i + e \cdot i^2 + 3 \cdot \sigma_{gpi}^{ipv}; j++$ **do**
▷ meet Eq. 2.31
14: **if** $I_{blob}(i,j) \neq 0$ **then**
15: ▶ calculate $G_{pi}(c,d,e,i,j)$ ▷ using Eq. 2.23 or Eq. 2.30
16: ▶ $p_{Gpi}(d,e) += w_{mke}(i,j) \cdot G_{pi}(c,d,e,i,j)$
17: **end if**
18: **end for**
19: **end for**
20: **end for**
21: **end for**
22: **end for**
23: ▶ find d_l^*, e_l^*, d_r^*, e_r^* from $p_{mke}^l(d,e)$ and $p_{mke}^r(d,e)$, using Eq. 2.33 and Eq. 2.34

In summary, the original multi-kernel method is enhanced in order to purchase a real-time ability. Three modifications are implemented: looking-up table to approximate kernels, discretizing $G_{pi}\left(x_i^{bev}, y_i^{bev}\right)$ to simplify computation, and limiting the range of $p_{mke}(d,e)$ to reduce calculation time.

The left and the right marking probabilities are computed together as in Eq. 2.32 to obtain the optimized parameters d_l^*, e_l^*, d_r^*, e_r^*, with a geometrical constraint presented in Eq. 2.33 and Eq. 2.34.

$$d_l^*, e_l^*, d_r^*, e_r^* = \underset{d_{j_l}, d_{j_r}, e_{j_l}, e_{j_r}}{\arg\max} \left[p_{mke}\left(d_{i_l}, e_{j_l}\right) + p_{mke}\left(d_{i_r}, e_{j_r}\right) \right]. \tag{2.32}$$

$$d_{j_r} \in \left(d_{j_l} - \triangle d_{para}, d_{j_l} + \triangle d_{para}\right), \tag{2.33}$$

$$e_{j_r} \in \left(e_{j_l} - \triangle e_{para}, e_{j_l} + \triangle e_{para}\right). \tag{2.34}$$

Therefore, the marking model parameters c_l^* and c_r^* are estimated in the initialization step, and d_l^*, e_l^*, d_r^*, e_r^* are optimized using multi-kernel method. The current lane markings are detected.

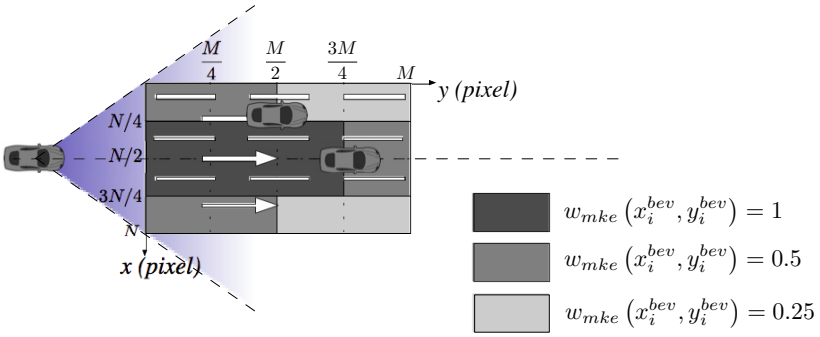


Figure 2.12: *Assignment of hierarchical weight* $w_{mke}\left(x_i^{bev}, y_i^{bev}\right)$ *in BEV space.*

2.4 Self-assessment Indicator

2.4.1 Indicator

When a marking in a frame is estimated, a self-assessment indicator is designed to qualify the detection. To this end, a confidence $conf_{ld}$ is employed to measure a detected marking. At first, a set Φ_{ld}^{conf} of pixels with higher contribution to the detected markings (c^*, d^*, e^*) are defined as follows:

$$\Phi_{ld}^{conf} = \left\{ \left(x_i^{bev}, y_i^{bev}\right) | G_{pi}\left(c^*, d^*, e^*, x_i^{bev}, y_i^{bev}\right) \geq Th_{G_{pi}} \right\}, \tag{2.35}$$

where $i = 1, ..., n_{ld}^{conf}$, $Th_{G_{pi}}$ is confidence threshold, n_{ld}^{conf} is the total number of high contributed pixels in the image.

Then the frame is divided into L_{conf} average blocks along y-axis. The set $\Phi_{l_{conf}}^{ld}$ of pixels in block l_{conf} is defined as:

$$\Phi_{l_{conf}}^{ld} = \left\{ \left(x_i^{bev}, y_i^{bev}\right) | \left(x_i^{bev}, y_i^{bev}\right) \in \Phi_{ld}^{conf}, \frac{M_{bev}}{L_{conf}} l_{conf} \leq y_i < \frac{M_{bev}}{L_{conf}} \left(l_{conf} + 1\right) \right\}, \tag{2.36}$$

where $l_{conf} = 0, 1, ..., L_{conf} - 1$, $i = 1, ..., n^{ld}_{l_{conf}}$, $n^{ld}_{l_{conf}}$ is the number of high contributed pixels in block l_{conf} .

When the numbers of high contributed pixels in all the blocks are derived, the confidence of a detection can be defined as:

$$conf_{ld} = \frac{n^{conf}_{ld}}{L_{conf}} \sum_{l_{conf}=0}^{L_{conf}-1} p^{ld}_{l_{conf}}, \tag{2.37}$$

where

$$p^{ld}_{l_{conf}} = \begin{cases} \frac{L_{conf} \cdot n^{ld}_{l_{conf}}}{n^{conf}_{ld}} & \left(0 \leq n^{ld}_{l_{conf}} < \frac{n^{conf}_{ld}}{L_{conf}}\right) \\ \frac{L_{conf} \cdot n^{ld}_{l_{conf}}}{(1-L_{conf}) \cdot n^{conf}_{ld}} + \frac{L_{conf}}{L_{conf}-1} & \left(\frac{n^{conf}_{ld}}{L_{conf}} \leq n^{ld}_{l_{conf}} < n^{conf}_{ld}\right) \end{cases}. \tag{2.38}$$

In confidence $conf_{ld}$, both the number of contributed pixels (n^{conf}_{ld}/L_{conf} in Eq. 2.37) and the distribution of these pixels ($\sum_{l_{conf}=0}^{L_{conf}-1} p^{ld}_{l_{conf}}$ in Eq. 2.37) are considered. A detection result which has more contributed pixels and whose contributed pixels are more equally distributed earns a higher confidence.

With confidence $conf_{ld}$ and an optimized threshold Th^*_{ld}, the detected markings can be classified. If both left and right detections are qualified, accept these two detections, and apply local parameter estimation in the next frame. If one detection is qualified while the other is not, accept the qualified detection and estimate the detection on the other side according to the qualified result, meanwhile apply local parameter estimation in the next frame. If neither of the detections is qualified, reject both of them and estimate the lane markings according to previous detections, and apply global parameter estimation in the next frame.

2.4.2 Ground Truth from KITTI Database

The lack of a standard and unified evaluation environment is one of the most critical problems in quantifying the performance of marking detection, as stated in Hillel et al. (2014). To assess our result in a public dataset and provide a universal evaluation for comparison, we make use of the road surface ground truth frames included in KITTI database Geiger et al. (2013), enriched with a lane marking annotation presented below. The ground truth dataset addresses three different road types: urban unmarked (UU), urban marked two-way road (UM), and urban marked multi-lane road (UMM). UM images are chosen to evaluate our method because the boundaries of road surface benchmark in this type represent exactly the reference markings. The annotation procedure is illustrated in Fig. 2.13. When a benchmark image is input as Fig. 2.13a, the benchmark road surface is projected into BEV space as cyan area in Fig. 2.13b. Then the left and the right boundaries of the road surface are considered as a pair of

Table 2.1: *Definition of confusion matrix elements.*

		self-assessment indicator	
		$conf_{ld} \leq Th_{ld}$	$conf_{ld} > Th_{ld}$
with ground truth	$MAE > 4pxl$	TN	FP
	$MAE \leq 4pxl$	FN	TP

reference markings, which are shown as blue pixels in Fig. 2.13b. For contrast, a pair of markings detected by our method is depicted as red pixels in Fig. 2.13b as well. Fig. 2.13c illustrates the road surface benchmark (cyan), the reference markings (blue) and the detected markings (red) in perspective image. When road surface is occluded by objects (i.e. barriers) in some UM images, the road surface is annotated manually.

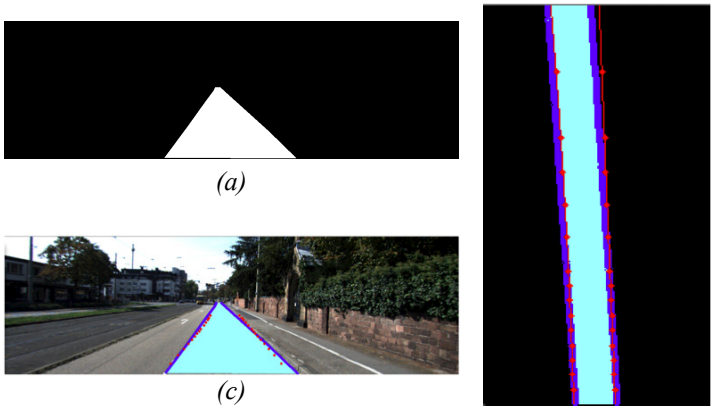


Figure 2.13: *The procedures from road surface benchmarks to lane marking benchmarks. (a) Road surface benchmark in perspective space; (b) road surface benchmark (cyan), reference markings (blue) and the markings (red) in BEV space; (c) road surface benchmark (cyan), reference markings (blue) and detected markings (red) in perspective space.*

2.4.3 Indicator Threshold Estimation

One usage of the reference markings is to derive the optimized indicator threshold T^*_{ld}. To this end, a confusion matrix is constructed as Table 2.1, including the required confusion matrix elements: true positive (TP), false positive (FP), true negative (TN) or false negative (FN).

After that, True Positive Rate $TPR = \frac{TP}{TP+FN}$ and False Positive Rate $FPR = \frac{FP}{FP+TN}$ with different T_{ld} are calculated, shown as the green stars in Fig. 2.14. A function $y(x) = \left((a_{roc})^x - (b_{roc})^x\right) / \left(a_{roc} - b_{roc}\right)$ is fitted as Receiver Operating Characteristic (ROC) curve Fawcett (2006) according to the sample points, depicted as red curve in Fig. 2.14. The optimized point is obtained as the nearest point on ROC curve to the perfect classification point $(0, 1)$ Fawcett (2006), depicted as the red dot in Fig. 2.14. Optimized threshold of self-assessment indicator Th^*_{ld} is estimated corresponding to the optimized point. This estimation process is executed offline.

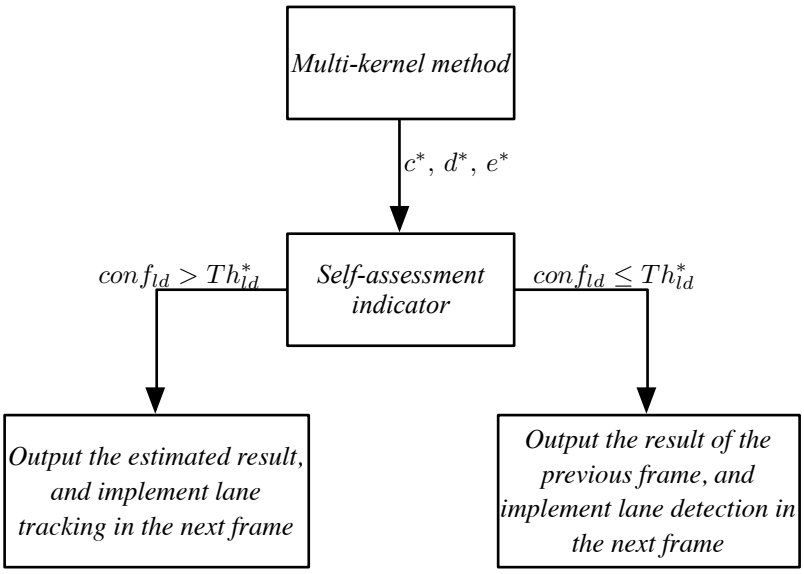


Figure 2.15: *Indicator based lane detection strategy.*

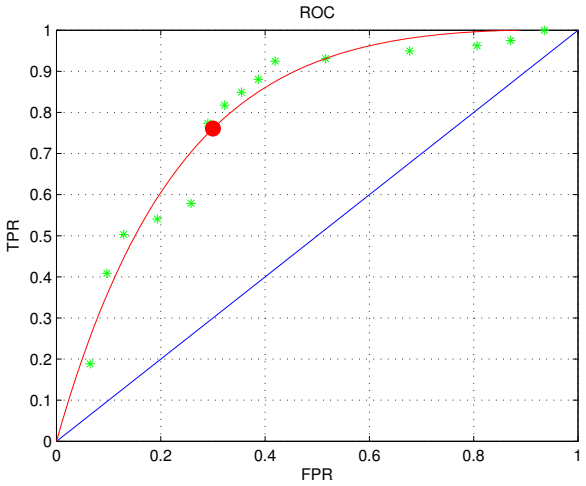


Figure 2.14: *ROC curve. Green stars are samples, red curve is the fitted exponential function, and the red dot is the optimized point.*

Therefore, the strategy of the lane detection project is constructed, depicted as Fig. 2.15. Multi-kernel based method estimates the marking model parameters c_l^*, d_l^*, e_l^*, c_r^*, d_r^*, e_r^*. The confidence $conf_{ld}$ of a single lane marking can be derived using self-assessment indicator. If the sum of left and right confidences is greater than indicator threshold Th_{ld}^*, it is implied that the achieved lane markings are qualified, the marking models will be produced as output results, and in the next frame lane tracking strategy is carried out. Otherwise, the markings are considered as poor detection, these two marking models from MKE method are rejected. The marking results of previous frame are given as the output results of this frame. The camera stream we used in this thesis has the frequency of 0.1s, so the model parameters vary little. The marking models of previous frame are preferable to a pair of bad detection. In this case, the MKE algorithm will run with the whole parameter range, so as to search for a qualified

detection.

2.5 Experimental Validation

2.5.1 Results of Proposed Method

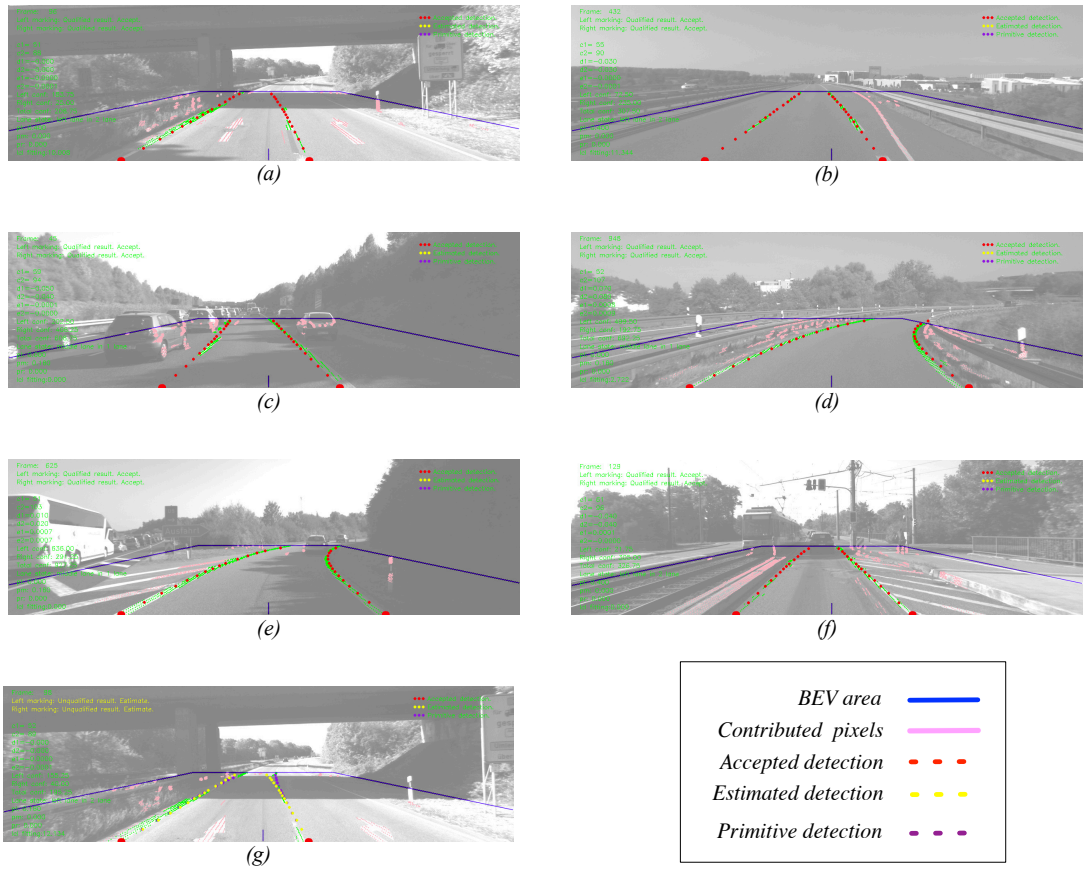


Figure 2.16: *Example detection results within various outdoor conditions. (a) Tunnel entrance with strong light variation; (b) discontinuous markings; (c) heavy traffics; (d)curves; (e) curves and shadows; (f) irregular shadows and polluted road surface; (g) estimated markings according to confidence indicator.*

This algorithm has been tested on 12 different challenging scenarios in KITTI database Geiger et al. (2013), including more than 1900 frames. Fig. 2.16 shows several detected lane markings in various outside conditions. The image processing area is marked within blue lines. The pink pixels are the lane marking candidates after image processing. The green pixels are the corresponding high contributed pixels in Φ_{ld}^{conf} from Eq. 2.35. The detected lane markings are depicted as red dots in the image processing area. In Fig. 2.16a, the number of candidate pixels for the right lane is limited because of the strong contrast near tunnel entrance, but the constraint of parallel markings ensures both detections. Fig. 2.16b shows the detection of dashed markings, which is more difficult than continuous marking detection in practice. Marking initialization step helps this detection, because the zero order components of marking model

are estimated before multi-kernel fitting. The detection in heavy traffic conditions is shown as Fig. 2.16c, thanks to the cell-based blob algorithm, numerous outlier pixels caused by vehicles are eliminated. Fig. 2.16d depicts the detection result in a corner situation, the parabolic model fits the markings. Fig. 2.16e-f are more complicated road situations including curves, multi markings, heavy traffic or irregular shadows, our approach derives detection results in these environments. Fig. 2.16g shows a special condition, the primitive result (purple dots) is detected, but this result is rejected by self-assessment indicator, a pair of estimated markings (yellow dots) are adopted as output result. The results above perform the marking detection ability of the proposed method in highway and road scenes.

Fig. 2.17 presents the time cost of the proposed algorithm on three scenarios of KITTI. The blue curves are the time cost, while the red lines represent a lane detection state in the current frame. Normally, the algorithm is in lane tracking state. If an unsuccessful detection is produced, the algorithm turns to lane detection state. In Fig. 2.17, totally 2840 frames are tested, all of which are real road situations. The average time cost is 21.6 ms. About 96.65% of the frames adopt local parameter estimation (7 scales for both model parameter d and e), which is lane tracking states. About 3.35% of the frames apply global parameter estimation (40 scales for both model parameter d and e), nominated as lane detection states.

In Fig. 2.17a, the average time cost is 20.73 ms, with the maximum time 66.82 ms and minimum time 14.72 ms. 94.91% of the frames are in lane tracking state (1062 frames in lane tracking, 57 frames in lane detection). The average time of tracking state is 19.96 ms, while that of detection state is 35.20 ms. In Fig. 2.17b, average time is 21.90 ms, and the maximum time is 63.45 ms, minimum time is 17.03 ms. Among 836 frames, 824 frames in lane tracking, 12 frames in lane detection, which means 98.56% of the frames are in lane tracking state. The average time of tracking state is 21.81 ms, while that of detection state is 28.29 ms. From Fig. 2.17c, we can observe that the average time cost is 22.40 ms, with the maximum time 56.03 ms and minimum time 16.15 ms. In this scenario, 859 frames in lane tracking, 26 frames in lane detection, which means 97.06% of the frames are in lane tracking state. The average time of tracking state is 22.16 ms, while that of detection state is 30.21 ms. The lane detection state takes more time because more parameter candidates are considered in SHT. This type of state usually happens in the frames where the markings are difficult to fit. For instance, there is a tunnel entrance with strong illumination difference (around frame 100 in Fig. 2.17a and frame 1100 in Fig. 2.17a), there are no markings in certain sections (around frame 700 in Fig. 2.17b), the rover is making a turn at road intersection (after frame 900 in Fig. 2.17c). According to the time cycle of KITTI database frames (100ms), the proposed method (average time is 21.6 ms) is adequate to process the frame streams in real-time.

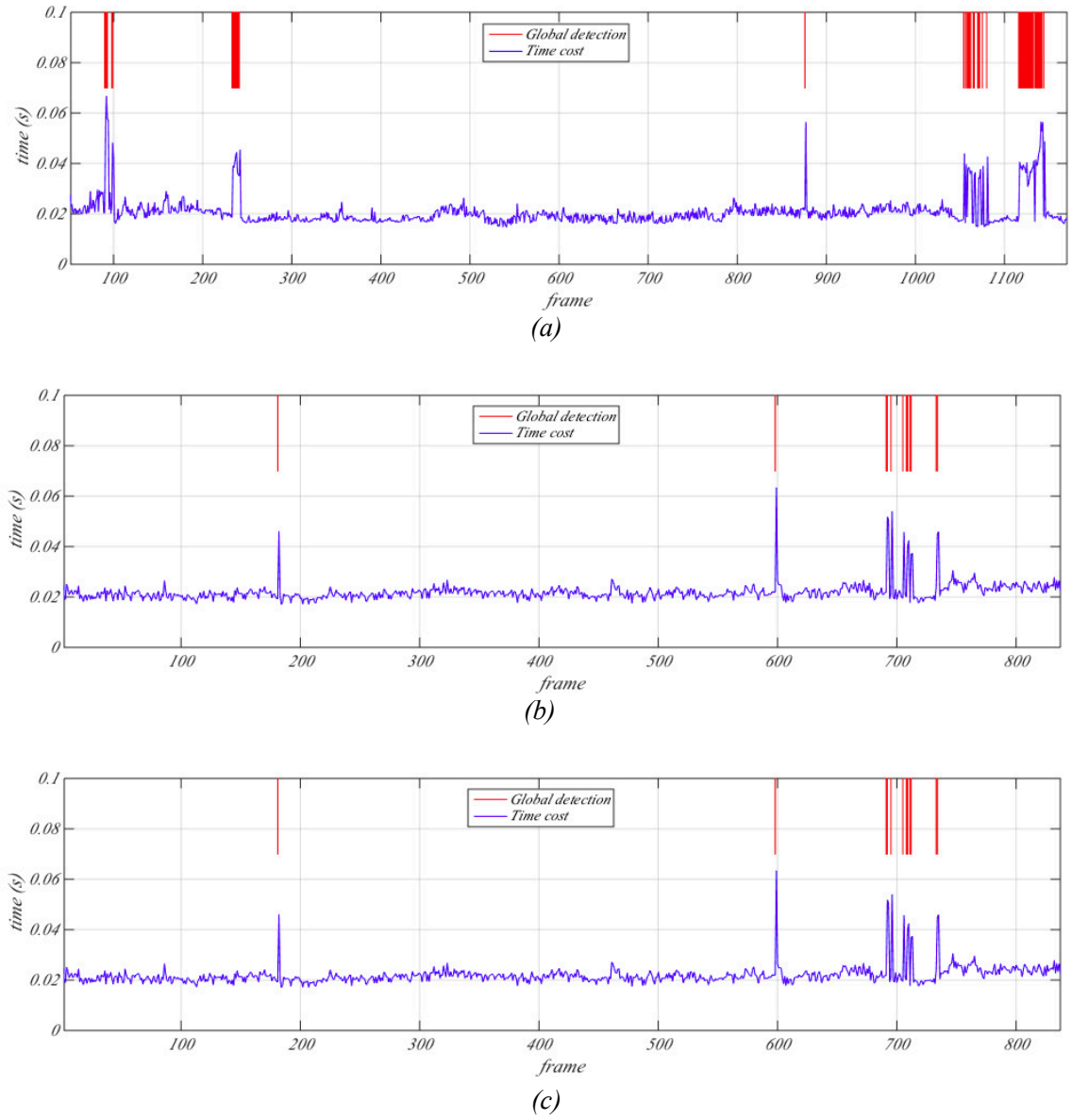


Figure 2.17: *Time cost of lane detection algorithm. (a)-(c) represent three different scenarios of KITTI database. Blue curves are time cost, red curves are lane tracking/detecting state.*

Besides real time ability, the detection accuracy and precision are also key parts in lane marking detection. To obtain the indexes such as Mean Absolute Error (MAE) and success ratio, the benchmark markings are required. However, the KITTI database only provides several ground truth frames for road surface detection. The KITTI benchmark information can be processed to lane marking ground truth, and thus can evaluate lane marking detection result. The results of this part of work will be introduced in the next subsection. Here the evaluation method according to lane width is represented. When the left and right lane marking models are estimated, the lane width can be computed according to Eq. 2.13. Normally, the width of a lane keeps stable for a relative long distance. If a peak appears in the lane width curve, the detection result is obviously unreasonable. Blue curves in Fig. 2.18 show the detected lane width of two road scenarios in KITTI. In Fig. 2.18a, the lane width has risen from 35 pixels (3.5 meter) to 50 pixels (5 meters), between frame 900 to 1100, because the rover drives to an on-ramp road, whose lane width is wider than ordinary lane width. Besides, the lane width of frame 1135-1145 occurs a peak, which increases from 35 pixels (3.5 meter) to 50 pixels (5 meters) and decreases back to 35 pixels (3.5 meter) in about 10 frames (1 second). A lane width for only 10 frames is unusual. In fact, the algorithm has been affected by strong light difference and poor detections are produced as output. Similarly, the proposed method estimates error detections in frame 827-835 of Fig. 2.18b, a peak appears in these frames. To have quantified result, the benchmark lane width is marked as red curves in Fig. 2.18. An error detection is defined when the difference between estimated width and benchmark is equal to or greater than 5 pixels (0.5m). In Fig. 2.18a, 35 error detections are found among 1119 frame. In Fig. 2.18b, an amount of 69 error detections is discovered from 836 frames. The total success ratio according to lane width is 94.68%. From the lane width aspect of view, the lane detection algorithm owns a high accuracy.

2.5.2 Results of Methods Comparison

The experiment results with benchmarks provided by KITTI are presented in this part. To further support our method, several algorithms are introduced for comparison, including the techniques from other papers and MKE methods with different kernels. Three lane detection algorithms, Method A Aly (2008), Method B Liu et al. (2011) and Method C Li et al. (2013), are introduced to compare with the proposed method. In Method A, selective oriented Gaussian filters are applied as image processing strategy. RANSAC is then used to fit Bezier Splines. All the lane markings in the frame are detected. Method A and our method use totally different marking models and fitting methods. The contrast of these two methods is regarded as the comparison between two different frameworks. In Method B, a multi-kernel based framework is introduced to determine parameters of a parabolic model. Method B and our method have the same

multi-kernel estimation framework. Different from Method B, our method proposes a parametric Gaussian initialization step, and hierarchical weights. The comparison between Method B and our method can imply the effect of the improvements in our methods. Method C employs Hough transform to fit straight lines. The key advantage of this method is high real-time ability. In tradition, the estimated markings are less precise than complex models.

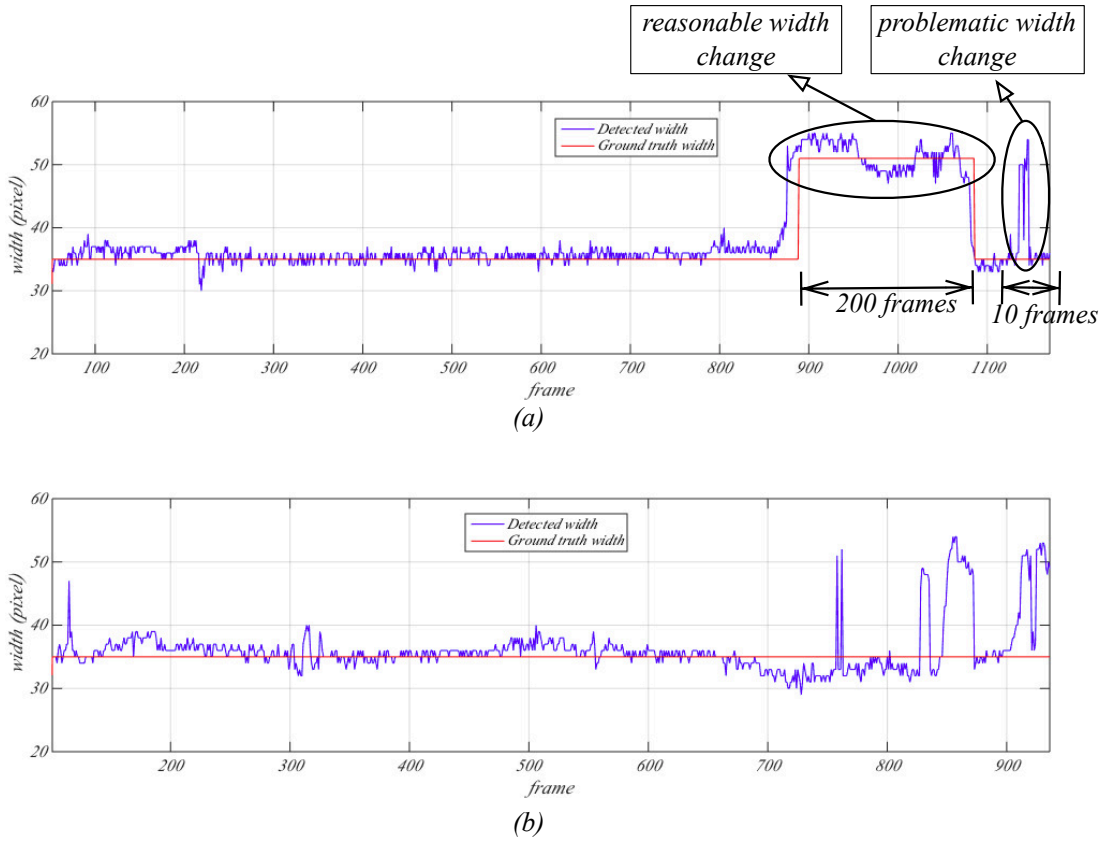


Figure 2.18: *Lane width of two KITTI scenarios. (a) and (b) are two scenarios in KITTI. Blue curves are estimated width curves, red ones are manually made ground truth.*

Besides, the simulation results of MKE methods with different kernels are analyzed. The original Gaussian kernel is shown the as the pink curve in Fig. 2.19. To simplify the model, several approximations are introduced, for instance, triangle kernel (blue lines in Fig. 2.19) in Eq. 2.28 and Eq. 2.29, Taylor expansion of Gaussian kernel (cyan dashed lines in Fig. 2.19) from Eq. 2.39 and Eq. 2.40, and looking-up table (red dots in Fig. 2.19) as Eq. 2.41 and Eq. 2.42.

The Taylor expansion kernel is given as:

$$K_x^{tay} = \begin{cases} \frac{4\cdot\left(\sigma_x^{ori}\right)^4 - 2\cdot\left(\sigma_x^{ori}\right)^2\cdot\left(x-x_i^{bev}\right)^2 + \left(x-x_i^{bev}\right)^4}{4\cdot\left(\sigma_x^{ori}\right)^4\cdot\sqrt{2\pi\sigma_x^{ori}}}, & x \in \left[x_i^{bev} - \sigma_x^{ori}, x_i^{bev} + \sigma_x^{ori}\right] \\ 0, \quad else \end{cases}, \quad (2.39)$$

$$K_y^{tay} = \begin{cases} \frac{4\cdot\left(\sigma_y^{ori}\right)^4 - 2\cdot\left(\sigma_y^{ori}\right)^2\cdot(y-y_i^{bev})^2 + (y-y_i^{bev})^4}{4\cdot\left(\sigma_y^{ori}\right)^4\cdot\sqrt{2\pi\sigma_y^{ori}}}, & y \in \left[y_i^{bev} - \sigma_y^{ori}, y_i^{bev} + \sigma_y^{ori}\right] \\ 0, \quad else \end{cases}, \quad (2.40)$$

where σ_x^{ori} and σ_y^{ori} are the variance of original Gaussian kernel, $x = c + dy + ey^2$. Looking-up table kernel model is represented as:

$$K_x^{tab}(z\Delta T_{mke}) = \frac{1}{\sqrt{2\pi\left(\sigma_x^{ori}\right)^2}} exp\left(-\frac{(z\Delta T_{mke} - x_i^{bev})^2}{2\left(\sigma_x^{ori}\right)^2}\right), \quad (2.41)$$

$$K_y^{tab}(z\Delta T_{mke}) = \frac{1}{\sqrt{2\pi\left(\sigma_y^{ori}\right)^2}} exp\left(-\frac{(z\Delta T_{mke} - y_i^{bev})^2}{2\left(\sigma_y^{ori}\right)^2}\right), \quad (2.42)$$

where $z \in \mathbb{Z}$, and ΔT_{mke} is the sampling interval.

The source codes of Method A and Method C is provided by the authors. Only the IPM parameters of KITTI experiment environment are set before the programs run. The code for Method B is programed according to Liu et al. (2011). Different kernels are packed as the forms of functions to add into MKE method. Source codes of all the methods are run using KITTI benchmark images. Obtained markings from detection methods are compared to reference markings, which are obtained from 95 benchmark images (190 benchmark markings) in 2.4.2. Comparison results of these methods are reported in Table 2.2, where **Time** is the run time per frame, **Correct** is the number of correct markings (a marking's Mean Absolute Error (MAE) is less than 4 pixels in BEV space is defined as a correct detection), **MAE** means the MAE of all the detected markings, **RMSE** is the root-mean-square error of the detections, and **Success ratio** is the percentage of correct detections in all the detections.

The average run time of Method A is 0.0210s, which is close to the average time 0.02s mentioned in Aly (2008). Some error detections of Method A are markings from neighborhood lane or road barriers, rather than the current lane marking, the parametric Gaussian based initialization in our method efficiently solves this problem. While other error detections are just a part of a lane marking, i.e. a short line in a dashed marking, not the entire road line. So the success ratio of Method A is relative low. The contrast between Method A and MKE method shows that MKE method has a higher success ratio, but is slower than method A.

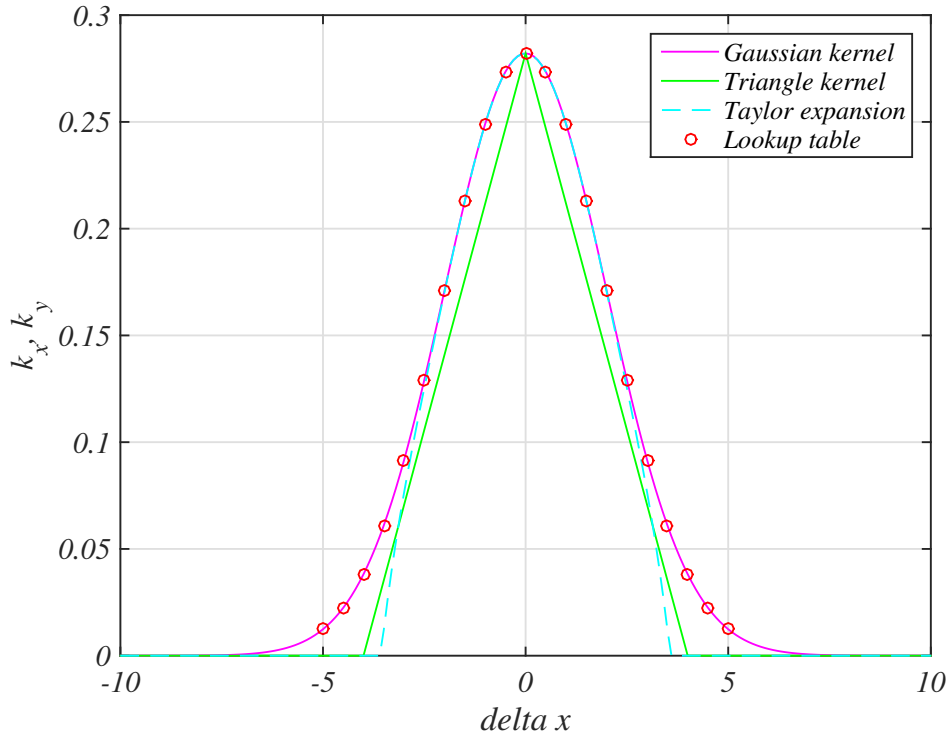


Figure 2.19: *Different kernels in MKE method. Pink: Gaussian kernel; Green: triangle kernel; cyan: Taylor expansion kernel; red dots: look-up table.*

Compared to Method B, MKE method has a less time cost. This comparison indicates that our improvement on original MKE method, for example, parametric Gaussian model initialization and the discretization of MKE descriptor, help to reduce average run time. In addition, the precision of MKE is about $10-15\%$ higher than that of Method B, because the parametric Gaussian based initialization step and hierarchical weights improve marking detection performance.

Original Hough transform for straight line estimation is exploited in Method C, the straight line model is simplest model in lane detection, so the processing time of Method C is the minimum among all the methods in the table. However, the straight line model cannot fit lane markings on real world perfectly, this is why the success ratio (62.11%) is lower than MKE method with parabola model. And in fitted markings, the MAE of Method C (1.93 pixel) is the highest in the table, testifying that the straight line model doesn't fit well the curved markings.

The results of MKE methods with four different kernels are raised as well. All three methods expect original Gaussian kernel based method take less time than original method, meanwhile the accuracy rates of these three methods are higher than that of the original kernel, which shows that the real time ability and the precision performance have been increased with the MKE method improvement. Among the three approximated kernel based methods, the MAE of look-up table kernel is the lowest, because the look-up table kernel extracts acquisition points from the original Gaussian

Table 2.2: *Methods comparison using KITTI dataset.*

	Time (ms)	Correct (marking)	MAE (pixel)	RMSE (pixel)	Success ratio(%)
Method A Aly (2008)	21.0	102	1.47	22.81	53.68
Method B Liu et al. (2011)	155.8	113	1.05	20.31	59.47
Method C Li et al. (2013)	13.5	118	1.93	7.17	62.11
MKE method (Gaussian kernel)	72.1	132	1.01	9.72	69.47
MKE method (Triangle kernel)	63.5	142	1.33	13.40	74.74
MKE method (Taylor expansion kernel)	67.7	142	1.39	13.32	74.74
MKE method (Look-up table kernel)	68.8	138	1.12	11.73	72.63

model. However, the computation time of this method is greater, since the addressing time spends more time than the kernel computation in my program. Between triangle kernel and Taylor expansion kernel methods, the success ratios are exactly the same, and are the highest of all the methods in the table. But triangle kernel takes less time than Taylor expansion kernel, because the model of triangle kernel is less complex. With the analysis, the triangle approximation kernel is selected as the kernel in lane detection method.

The time cost of our method in Table 2.2 is an average of global detection time, because the benchmark images are discontinuous road scenes, it is not feasible to introduce local lane marking detection. In addition, self-assessment indicator is not able to provide estimated detections in discontinuous frames. The performance of our method can be improved in continuous scenarios.

In addition, these four MKE methods are tested in KITTI database. The curves of computation time in 3 longest KITTI scenarios are shown in Fig. 2.20. Purple, green, cyan and red represent time cost of Gaussian kernel, triangle kernel, Taylor expansion kernel and look-up table kernel respectively. We can observe that the curves of the computation time of all the methods are similar from Fig. 2.20. Statistics corresponding to Fig. 2.20 is given in Table 2.3. The indexes of all the methods are similar, and the triangle kernel owns the least time cost.

2.6 Conclusion

This section demonstrates a real-time, reliable, and precise lane marking detection method applied in real traffic conditions, thanks to an improved multi-kernel based estimation method with hierarchical weights. A second-order derivative filter is proposed to extract lane marking pixels. The time cost is less than complex extraction method.

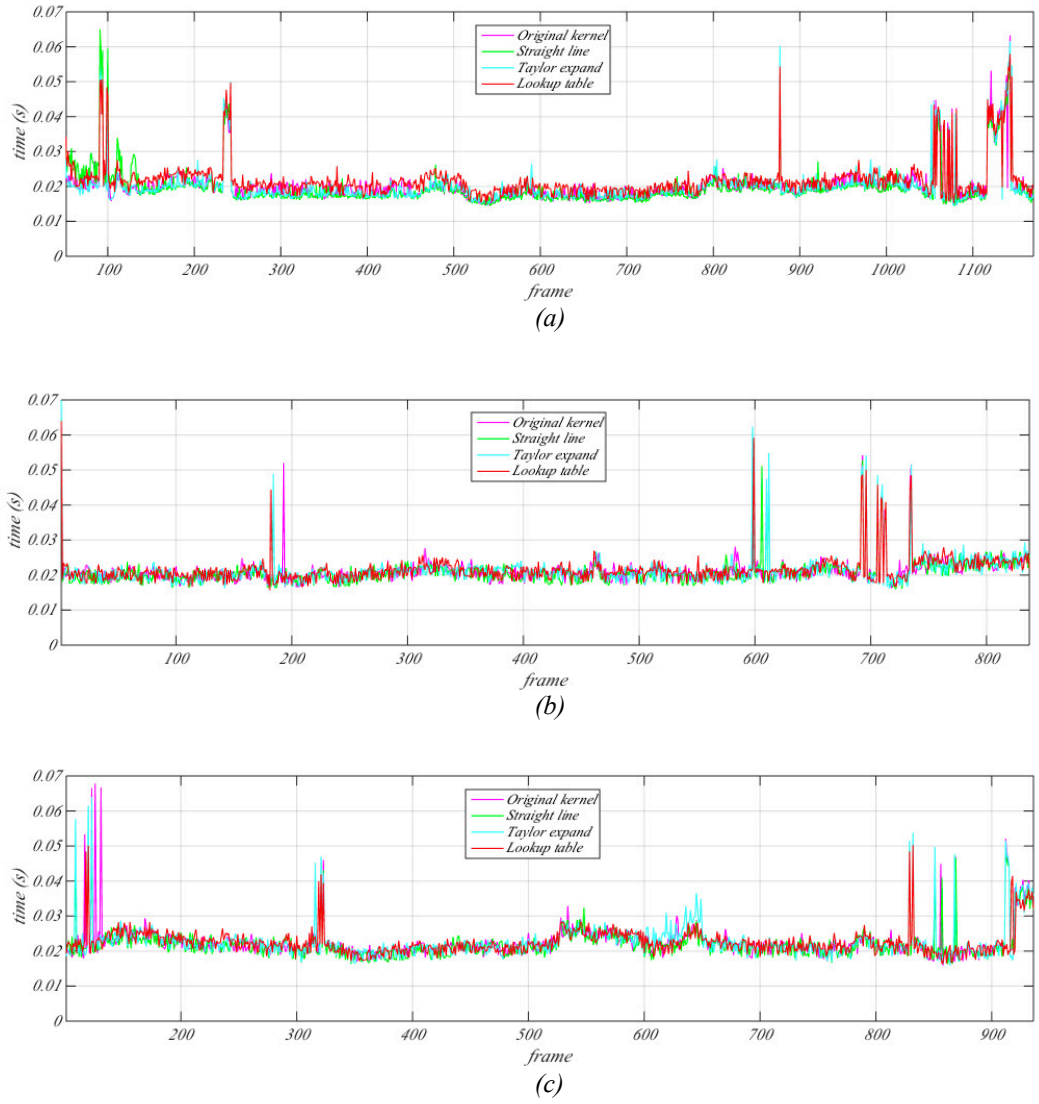


Figure 2.20: *Run time of four MKE methods in different KITTI scenarios. (a)-(c) are the algorithm run time of different kernels on different KITTI scenarios.*

Table 2.3: *Run time of four MKE methods in KITTI.*

	Gaussian kernel	Triangle kernel	Taylor expansion	Look-up table
Tracking frames	2686	2689	2693	2699
Tracking rate (%)	96.20	96.31	96.45	96.67
Maximum time (ms)	67.89	65.36	69.85	64.02
Minimum time (ms)	14.52	14.42	14.47	15.22
Average tracking time (ms)	20.79	20.37	20.79	21.54
Average detection time (ms)	32.10	32.52	33.19	33.24
Average total time (ms)	21.22	20.82	21.23	21.93

A cell-based blob algorithm is introduced to alleviate outlier pixels (i.e. heavy traffic, disturbed markings and shadows). A parametric Gaussian model based initialization is used to estimate zero order model parameters, which greatly reduces the time consuming of curve fitting. A multi-kernel based parameter estimation with hierarchical weights is recommended to obtain the rest model parameters. In the end, a confidence indicator is applied to improve the qualification of output markings.

Chapter 3

Marking Based Vehicle Localization

Contents

3.1 Introduction

In this section, the state-of-the-art on vehicle localization is surveyed according to the proposed methods. At first, stand-alone GPS localization approaches are examined. To address the existing problems of single GPS positioning, different information sources are employed, which are 1) vehicle motion sensors and vehicle models, 2) GIS. Navigation methods integrating these perception models are discussed, the advantages and disadvantages are analyzed. As one effective positioning approach, map-matching algorithms are classified into four categories: geometric approaches, topological techniques, probabilistic methods, and advanced algorithms, the strategies of each part is developed. To further improve the positioning ability, vision sensors are exploited as a novel information source. Road marking based localization is a clear example in the combination of digital maps and vision information. In the following our method will be stated and studied. In the end, our method is compared with other approaches, the contributions of this work will be demonstrated. The structure of the state-of-the-art is enlisted as Fig. 3.1.

3.1.1 Vehicle Localization

Vehicle localization plays a fundamental and critical role in Intelligent Transportation System (ITS) because it is the prior task for higher level operations, such as traffic flow control and vehicle network communication. Several methods have been proposed to improve the accuracy of the results using different technologies, a state-of-the-art is summarized in Skog and Handel (2009).

Stand-alone GPS. Before considering extra information, most of the localization methods Enge et al. (1996); Marais et al. (2005) exclusively rely on Global Navigation Satellite System (GNSS) signals. GPS receivers can compute the propagation time of signals, using the time-of-arrival of satellite signals and their transmission time. With the help of stationary GNSS receiver at a known location, the Differential GNSS (DGNSS) technology is applied to compensate the area restriction problem Huang and Tan (2006). Even if the receiver's positioning accuracy is enhanced Sunderhauf et al. (2013), many problems still remain. Buildings can cause multi-path phenomenon by blocking satellite signals; In urban environment, the multi-path propagation and outage situation impact positioning accuracy. So the GPS error is bounded error on current state, which is independent of previous GPS measurement.

Vehicle motion model. Localization methods based on only GPS signals are vulnerable, extra supports are introduced to obtain the expected accuracy, integrity, availability and continuity of service. The use of a vehicle motion model is an alternative to single GPS based localization. Dead Reckoning (DR) predicts relative movement of vehicle using courses and velocities from vehicle sensors. With previously estimated

positions, DR can observe the vehicle's current position. At first, gyroscope, compass, or differences in wheel speed measurements are used to determine the attitude or heading of the vehicle. Then, attitude or heading information is used to project the in-vehicle coordinates measured acceleration, velocity, or traveled distance. Finally, traveled distance, velocity, or acceleration is integrated over time to obtain position and velocity estimates in the navigation coordinate frame. In Obradovic et al. (2006), different positioning methods using DR are studied. As an alternative of DR, Inertial Navigation Systems (INSs) are able to produce not only vehicle position, but also the orientation and the vehicle velocity. In DR and INS systems, a bias in the accelerometer measurements causes error growth that is proportional to the square of the operation time, and a bias in the gyroscopes causes error growth that is proportional to the cube of the operation time El-Sheimy and Niu (2007); Tan and Park (2005). In other words, the cumulative error observed in odometry, remains the most critical problem of DR and INS sensors.

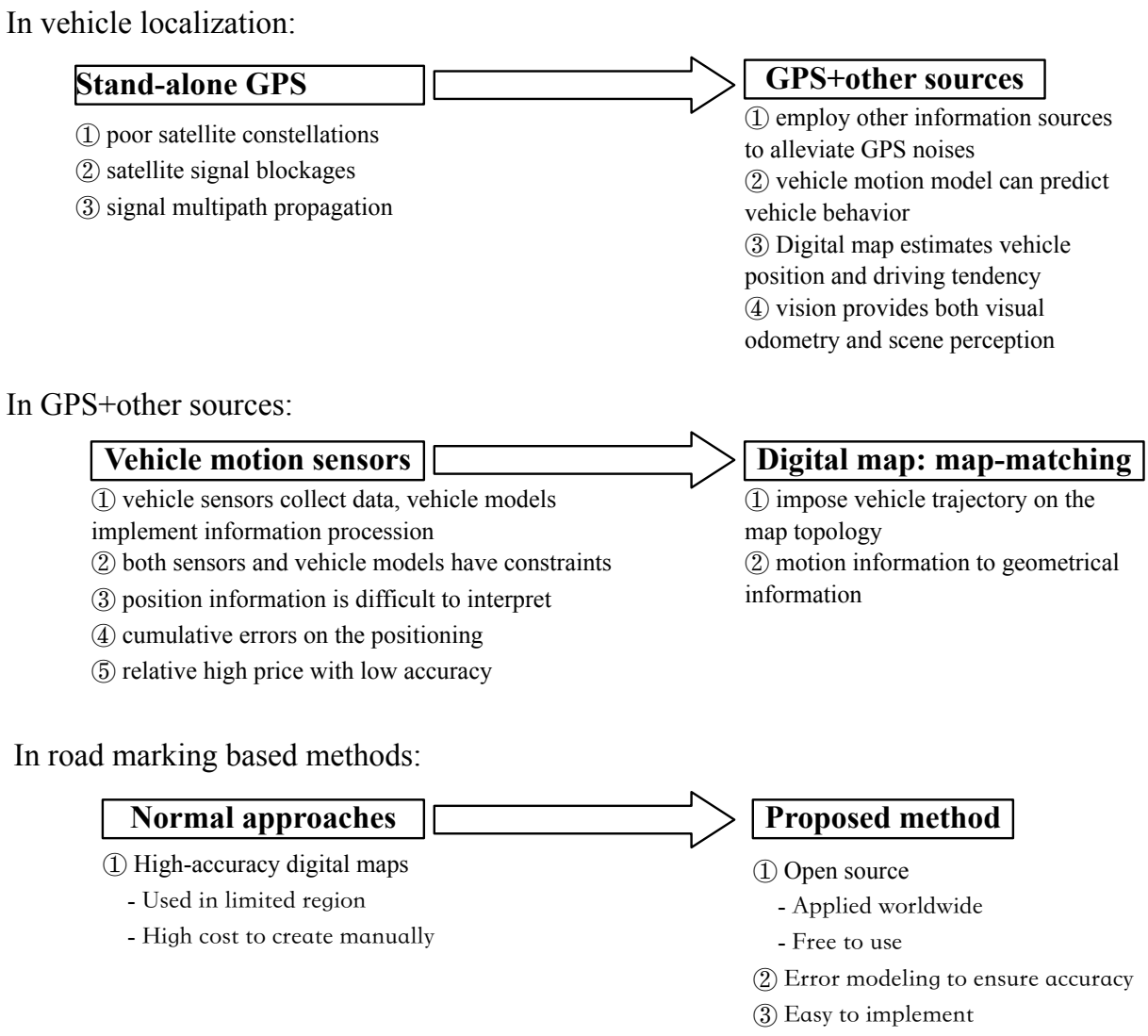


Figure 3.1: *Architecture of state-of-the-art in localization.*

Since the nature of these positioning techniques is different, distinct drawbacks are produced, such as bounded error from GPS, and cumulative error from DR/INS. The combination of the two systems compensates each other's error. INS/GPS sensor can be fused with a State-Dependent Riccati Equation (SDRE) filter Nemra and Aouf (2010), an Extended Kalman Filter (EKF) Redmill et al. (2001) or a particle filter Seo et al. (2013). In Amini et al. (2014), an integrated localization algorithm is represented that exploits all possible data from different resources including GPS, radio-frequency identification, vehicle-to-vehicle and vehicle-to-infrastructure communications, and dead reckoning.

Digital map: Map-matching algorithm. Normally, the trajectories of vehicles are restricted in the road area. Hence, a digital map of the road network is used to constraint the navigation solution of the in-car navigation system. This process is known as map matching Bernstein and Kornhauser (2000); Quddus et al. (2006a, 2007), and is detailed in the following subsections.

3.1.2 Map-matching Algorithm

Map-matching algorithms associate the positioning data from GPS sensors with Geographic Information Systems (GIS). By means of this process, it is possible to identify the current road section on which the host vehicle is traveling, and redefine the position of the host vehicle on the identified road section. The purpose of a map-matching method is to enhance the localization performance, and in advance to improve navigation function of ITS. In the following the state-of-the-art related to map-matching approaches are classified into the categories: geometric approaches, topological techniques, probabilistic methods, and advanced algorithms.

During the data association process, a geometric approach takes only the road section shapes into account, without the connection relationships of different links. The techniques of this category include matching a single vehicle position to a single map point Bentley and Maurer (1980), a single vehicle position to a map road curve White et al. (2000), and a vehicle trajectory curve to a map road curve Phuyal (2002). Euclidean distance is used to proceed point-to-point matching. In point-to-curve matching, the road curve is assumed as a poly-line, and the average distance between vehicle and all single lines of a road curve is used to determine map matching. Curve fitting methods can be introduced to deal with curve-to-curve matching problems.

Methods in this category own high real-time ability because these techniques have low complexity. However, geometric approaches are sensible to noises.

Topological approaches are able to solve the noise sensitivity problem occurred in geometric means, several "unstable" situations are discussed in Bernstein and Kornhauser (2000). The idea is to introduce topological information of map to help to reduce the

possible road curve candidates. In Meng (2006), an improved curve to curve matching algorithm is demonstrated. Several criteria such as Chen et al. (2003); Quddus et al. (2006b) are used to help identify the current road section, with the vehicle trajectory from GPS. Differences between vehicle bearing and road direction are computed, as well as distance to road segment, to determine the current road section. At road junctions, an algorithm is designed to decide which branch the vehicle drives to, considering connectivity, legal possible turn, and difference of vehicle bearing change and road direction changes.

Instead of a specific estimated location, an elliptical or rectangular confidence region is provided using probabilistic methods Quddus et al. (2007). In Ochieng et al. (2003), a confidence region is created according to Dead Reckoning position. This confidence region is computed only in complex conditions, such as junctions, reducing the algorithm executing time. In Ochieng et al. (2009), the elliptical error region is only constructed when the vehicle travels through a junction.

In recent years, advanced algorithms and methods are employed on map matching problem. On the one hand, algorithms with refined concepts are introduced. Common methods in vehicle localization include Kalman Filter Barth et al. (2013), EKF Tao et al. (2013a), optimization algorithm Lategahn et al. (2013), and methods relying on interval analysis Seignez et al. (2009). On the other hand, various cues are used to enhance map matching results. In Rodríguez F. et al. (2012), objects localization is regarded as one criterion to assist vehicle localization. In Wu and Ranganathan (2013), traffic signs (i.e. arrows, pedestrian crossings and markings) from vision are compared with the digital map to provide a transformation from vision space to map space.

3.1.3 Road Marking based Positioning

Road marking based localization approaches Publications as Wu and Ranganathan (2013); Gruyer et al. (2014) on this direction utilize high-accuracy digital maps, including the precise positions of the markings. In Wu and Ranganathan (2013), a transformation vector between vision space and map space are derived according to the features of traffic signs (i.e. markings and pedestrian crossings) captured via camera, which helps to adjust vehicle location. High-accuracy digital maps help to improve localization results. However, such maps are specially customized, which is time-consuming to fabricate and range-limited. In Tao et al. (2013b), a map of lane marking features is built. Vehicle localization is implemented using this map and vision based marking features.

In this chapter, a marking feature based vehicle localization method is proposed, using open source map and camera vision to adjust position signals from low-cost GPS receiver. The main idea is to extract and fit vision-based and map-based markings to optimize vehicle position. At first, a particle filter is designed to adjust rough GNSS

positions using lane marking evaluation model. This particle filter is intended to provide a first adjusting on rough vehicle positions. Then the road segments are estimated using multi-criterion estimation, the map-based lane markings of current state are then determined from an open source map database. After that, vehicle location is estimated according to integrating both vision-based lane markings and map-based lane markings, using a multi-kernel estimation method. Finally, potential errors in localization (e.g. map topology error) are located by probabilistic error modeling, so as to ensure the positioning accuracy of the proposed method. This work has been published in Lu et al. (2014b).

In our localization technique, low-cost GPS, open source map, and monocular camera are combined. Single GPS receiver brings bounded errors because the satellite signals may be blocked or reflected by buildings. The fusion of GPS and vehicle motion sensors (i.e. IMU) can relieve the GPS errors, but the high-cost motion sensors are not suitable for home use vehicles. Therefore, the combination of GPS and GIS is employed, because 1) map-matching algorithms are mature after development during decades, 2) vehicle trajectories are imposed on the map topology, alleviating GPS errors. When a camera position is fixed by GPS, this camera can provide local features to further increasing the precision of vehicle locations. Road marking is a proper local feature because this feature is familiar to drivers in road environment and can be detected with a higher confidence by camera. Many existing papers focusing on road marking based approaches exploit specialized map to obtain precise localization. However, these maps are costly to create, and their coverage area is limited. GIS database is applied in our method, the lane marking topology is constructed using the road properties in the map database. Therefore, low-cost GPS, open source map and camera are combined together to perform an accurate localization method.

3.2 Particle Filter

In a general discrete-time state-space model, the state of a system evolves according to:

$$\mathbf{x}_k = \mathbf{f}_k \left(\mathbf{x}_{k-1}, \mathbf{v}_{k-1} \right), \tag{3.1}$$

where $\mathbf{x}_k$ is a state vector at time k, $\mathbf{v}_{k-1}$ is the noise vector, $\mathbf{f}_k$ is the state evolution function. Information about $\mathbf{x}_k$ is only obtained through the observation $\mathbf{z}_k$, which is governed by:

$$\mathbf{z}_k = \mathbf{h}_k \left(\mathbf{x}_k, \mathbf{n}_k \right), \tag{3.2}$$

where $\mathbf{h}_k$ is a measurement function and $\mathbf{n}_k$ is the noise vector in measurement. We denote $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ as the transition probabilities represented by Eq. 3.1 and Eq. 3.2.

If $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$ are linear functions, and the noises $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ follow Gaussian distributions, the posterior distributions of $\mathbf{x}_k$ can be processed by Kalman filter Kalman and Bucy (1961). If these assumptions fail, particle filter is an alternative to solve the problem.

Particle filter is a sequential Monte Carlo methods based on particles representation of probability densities. A number of particles are created according to a certain probability distribution. Each particle represents one hypothesis of the states. Then these particles are propagated through the dynamic evolution model, which can be linear or nonlinear. In this way, a particle filter can handle any probability distributions. Measurement model is used to check how well a single particle fits the measurements. This fitting degree is represented by a weight. A high weight means the particle coincides well with the measurements. As a result, the particles and their weights present the probability distribution of the state variables. The use of Monte Carlo filtering methods can restrospect to Handschin and Mayne (1969); Handschin (1970), in which Monte Carlo method is used to obtain statistics moments of the posterior probability. A literature review West and Harrison (1997) expatiates on particle filter and other similar filtering methods, such as EKF Jazwinski (2007); Anderson and Moore (2012), Gaussian sum filter Sorenson and Alspach (1971) and the approximate grid-based methods Bucy and Senne (1971).

3.2.1 The Basis of Particle Filter

Particle filter aims to observe the posterior distribution of states $p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k})$ and the filtering distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ recursively in time. The distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is derived from Bayesian theory:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) \cdot p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1}, \tag{3.3}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{z}_{1:k-1}). \tag{3.4}$$

Generally, the integral of Eq. 3.3 cannot be carried out analytically, the solution is to approximate the intractable integrals by the means of a Monte Carlo sampling.

It is assumed that in particle filter is that the Probability Density Function (PDF) of state $\mathbf{x}_k$ is estimated by a group of weighted particles. Assume that at time $k-1$, a set of particles with the corresponding weights are created as $\left\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_{i=1}^{N}$, thus the posterior distribution of $\mathbf{x}_{k-1}$ is approximated as follows:

$$\hat{p}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \cdot \delta\left(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(i)}\right), \tag{3.5}$$

where $\delta(\cdot)$ is the Dirac-Delta function. Substituting Eq. 3.5 into Eq. 3.3 and Eq. 3.4, the distribution at time k is given as:

$$\hat{p}\left(\mathbf{x}_k|\mathbf{z}_{1:k-1}\right) = \sum_{i=1}^{N} p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right) \cdot w_{k-1}^{(i)}, \tag{3.6}$$

$$\hat{p}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) \propto \sum_{i=1}^{N} p\left(\mathbf{z}_k|\mathbf{x}_k\right) \cdot p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \cdot w_{k-1}^{(i)}. \tag{3.7}$$

In Eq. 3.7, a set of particles provide only one approximation in one iteration. The solution is to create particles $\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$ according to transitional probability:

$$p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right) \quad i = 1, ..., N, \tag{3.8}$$

so the particle weight is updated from the previous weight as:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \cdot p\left(\mathbf{z}_k|\mathbf{x}_k^{(i)}\right), \tag{3.9}$$

where $\sum_{i=1}^{N} w_k^{(i)} = 1$. Now the calculation of $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is sequential computation. Similarly, the joint posterior distribution $\hat{p}\left(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1}\right)$ is approximated as:

$$\hat{p}\left(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1}\right) = \sum_{i=1}^{N} w_{k-1}^{(i)} \cdot \delta\left(\mathbf{x}_{1:k-1} - \mathbf{x}_{1:k-1}^{(i)}\right). \tag{3.10}$$

3.2.2 Sequential importance Sampling (SIS)

Sequential Importance Sampling (SIS) is a Monte Carlo method addressing the prediction and update step of a particle filter, proposed in Liu and Chen (1995) at first. The purpose of SIS is to estimate the posterior distribution $p\left(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}\right)$ with N particles $\left\{\mathbf{x}_{0:k-1}, w_{k-1}^{(i)}\right\}_{i=1}^{N}$, and update these particles to derive an approximation to the latest posterior distribution $p\left(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}\right)$. The update step of SIS is carried out by the following equations:

$$\mathbf{x}_k^{(i)} \sim q\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, \mathbf{z}_k\right), \tag{3.11}$$

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^{(j)}\right) \cdot p\left(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}\right)}{q\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, \mathbf{z}_k\right)}, \tag{3.12}$$

where $q\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, \mathbf{z}_k\right)$ is the proposal distribution of importance sampling. The SIS filter procedure is outlined hereafter:

It is worth nothing that SIS approach is missing resampling step. The algorithm will inevitably suffer the problem of sample degeneracy, where only a few of the particles will have a significant weight, and all the other particles will have very small weights. A remedy to this problem is to introduce Effective Sample Size (ESS). ESS represents

how large a simple random sample from a target distribution would be required to estimate the function of interest in the importance sampling, which is used to measure the efficiency of importance sampling Liu (1996); Neal (2001). In SIS, resampling is implemented when the ESS falls below a certain threshold typically.

Algorithm 3.2.3 SIS filter

Input: - Measurements $\mathbf{z}_{0:k-1}$

Output: Observations: $\mathbf{x}_{0:k-1}$

▷ Initialization step

if $k = 0$ **then**
  for $i = 1; i < N; i++$ **do**
    ▶ Sample $\mathbf{x}_0^{(i)} \sim q(\mathbf{x}_0|\mathbf{z}_0)$
    ▶ Value $w_0^{(i)} = p\left(\mathbf{x}_0^{(i)}|\mathbf{z}_0\right) / q\left(\mathbf{x}_0^{(i)}|\mathbf{z}_0\right)$
  end for
end if

▷ Importance sampling

if $k > 0$ **then**
  ▶ Assume there are N particles $\left\{\mathbf{x}_{0:k-1}, w_{k-1}^{(i)}\right\}_{i=1}^{N}$
  for $i = 1; i < N; i++$ **do**
    ▶ Sample $\mathbf{x}_k^{(i)}$ according to Eq. 3.11
    ▶ Sample $w_k^{(i)}$ according to Eq. 3.12
  end for
end if
▶ $k = k + 1$

3.2.3 Sampling Importance Resampling (SIR)

Sampling Importance Resampling (SIR) algorithm is a variant of SIS Orhan (2012). The former is first demonstrated in Gordon et al. (1993), and developed independently in Kitagawa (1996); Isard and Blake (1996). The innovation of such filter is that a resampling step is introduced at every time to overcome the degeneracy problem.

If the proposal distribution $q\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, \mathbf{z}_k\right)$ in Eq. 3.13 is replaced by the state transition distribution $p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right)$, the SIS filter becomes the SIR filter without resampling step. If the resampling step is applied at every iteration, Eq. 3.11 and Eq. 3.12 are simplified to:

$$\mathbf{x}_k^{(i)} \sim p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right), \tag{3.13}$$

$$w_k^{(i)} \propto p\left(\mathbf{z}_k|\mathbf{x}_k^{(j)}\right), \tag{3.14}$$

SIR filter is implemented.

The key advantage of the SIR is its low complexity, since it only requires sampling from the distribution $p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right)$ and evaluating $p\left(\mathbf{z}_k|\mathbf{x}_k^{(j)}\right)$. The disadvantage is the independence of the proposal distribution $p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}\right)$ from the observations $\mathbf{z}_k$, which means that the states are refreshed without observation information directly). Besides, the sample impoverishment problem can be severe due to resampling at every iteration. However, the cost of SIR to pay for the convenience is expensive. The proposal density function ignores the information of the observations, which makes the filter inefficient and sensitive to outliers. Multinomial sampling can add substantial Monte Carlo variation to the algorithm.

In our lane marking based particle filter, SIR filter is adopted to estimate vehicle fixes, including initialization, prediction, updating, and SIR resampling. Map topology is exploited into our filter to help to determine vehicle velocity and bearing. So before explaining the marking based particle filter, the structure reconfiguration of OSM is elaborated.

3.3 Map Reconfiguration

In this section, a "lane marking" map topology is constructed. Different from the construction of lane marking based maps using EKF in Esparza-Jimenez et al. (2014), an open source database is introduced to create map topology. At first, the map organization composed of "nodes" and "ways" is derived from map data. Then the "node-way" topology is reconfigured to a serial of "road segments". The "segments" are transformed again to "cells". A cell represents a direct road lane in highway environment, and thus the left and right boundaries are lane markings of the lane. Therefore, a "lane marking" topology is achieved. The transmissions from "way" and "node" to "segment" and "cell" are depicted as Fig. 3.2.

3.3.1 Node and way

Map-based lane markings are obtained from OpenStreetMap (OSM), a collaborative project to provide open source map database. The basic components of OSM are a "node" and a "way". A node is one of the core elements in the OpenStreetMap data model, which represents a specific point in map space. It consists of a single point in space defined by its latitude, longitude and node id. According to the requirement of lane marking based localization, a node in ENU space is defined as:

$$nod_m = \left\{id_m^{nod}, lat_m^{nod}, lon_m^{nod}, x_m^{nod}, y_m^{nod}\right\}, \tag{3.15}$$

where id_n is a unique identification number provided by OSM, lat and lon are the latitude and longitude coordinate in an ECEF space, x_{enu} and y_{enu} are the corresponding

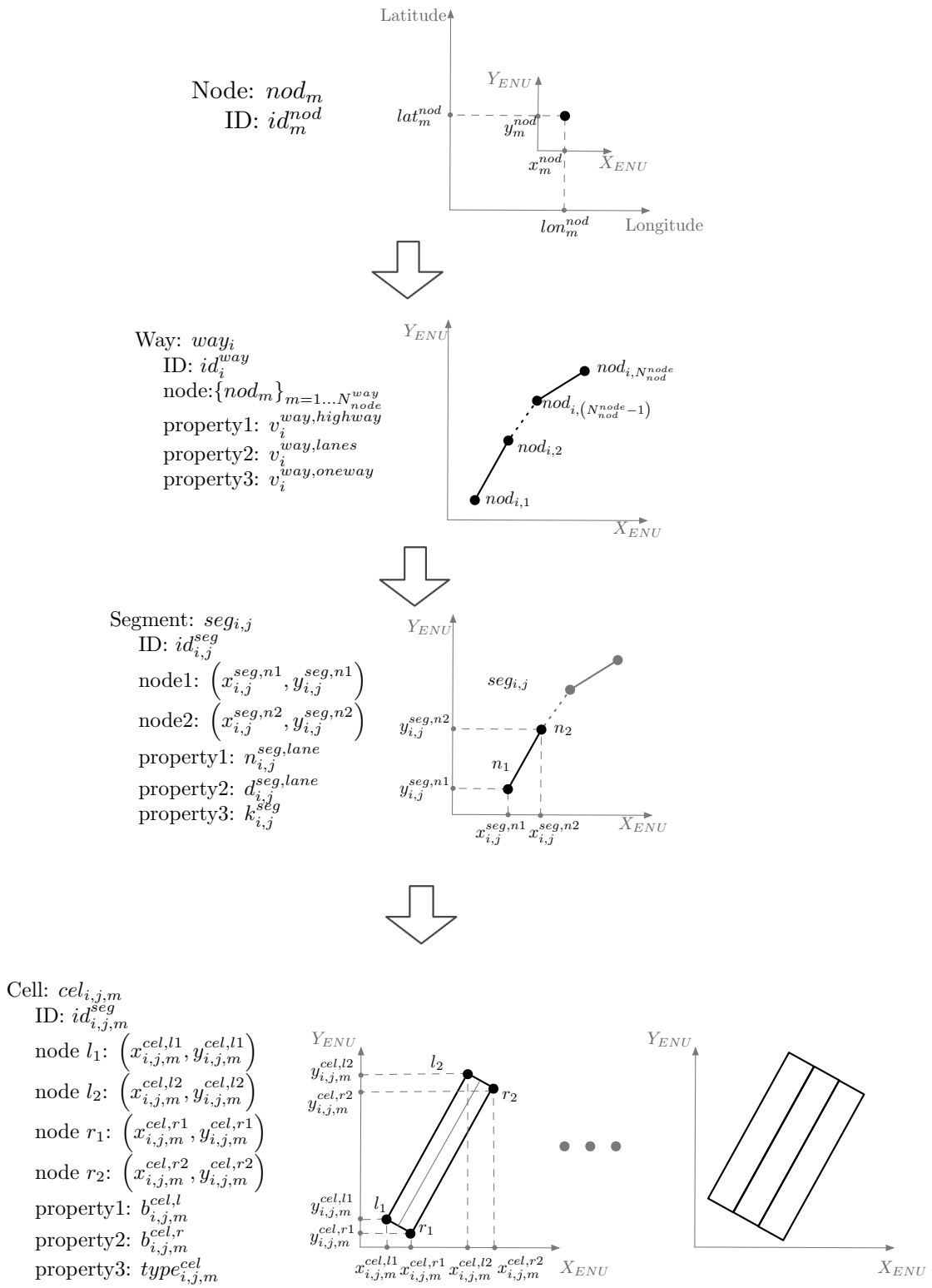


Figure 3.2: *Relationship among "way", "node", "segment" and "cell".*

coordinate in ENU space. In OSM data model, *lat* and *lon* are specified in degrees, resulting in a worst case a precision of $\pm 1cm$.

A way, the other basic unit in OSM, is able to denote an object in the map. A way is an ordered list of nodes, representing roads and other objects on the map. A way can be open or closed. A closed way is one whose last node on the way is also the first on that way, such as a road, a building, or a landuse. In the literature, a way refers specifically to a stretch of road, which is defined as:

$$way_i = \left\{ id_i^{way}, \{nod_m\}_{m=1...N_{node}^{way}}, v_i^{way,highway}, v_i^{way,lanes}, v_i^{way,oneway} \right\}, \tag{3.16}$$

where id_i^{way} is the id number derived from OSM database. $\{nod_m\}_{m=1...N_{node}^{way}}$ is a list of nodes to form a part of road as way definition, nod_m includes all the elements contained in a node. $v_i^{way,highway}$, $v_i^{way,lanes}$ and $v_i^{way,oneway}$ are three properties of a way, provided by OSM. $v_i^{way,highway}$ represents the highway type. For instance, $v_i^{way,highway} = motorway$ means a restricted access major divided highway, $v_i^{way,highway} = motorway\_link$ represents that the link roads (slip roads/ramps) leading to/from a motorway from/to a motorway or lower class highway, $v_i^{way,highway} = trunk$ means the most important roads in a country's system that aren't motorways, $v_i^{way,highway} = trunk\_link$ are the link roads (slip roads/ramps) leading to/from a trunk road from/to a trunk road or lower class highway, and $v_i^{way,highway} = primary$ represents the next most important roads in a country's system. $v_i^{way,lanes}$ is the lane number of the current way. $v_i^{way,oneway}$ is used to judge if this way is a one direction or both directions.

3.3.2 Segment

To derive the marking topology, the concepts "node" and "way" from OSM are not adequate. To convert way properties of OSM into visualized geometrical parameters, a "segment" is defined as a straight line section of a poly-line "way". A way contains one or more segments, as a straight line or a poly-line. The j^{th} segment in i^{th} way is defined as:

$$seg_{i,j} = \left\{ id_{i,j}^{seg}, x_{i,j}^{seg,n1}, y_{i,j}^{seg,n1}, x_{i,j}^{seg,n2}, y_{i,j}^{seg,n2}, n_{i,j}^{seg,lane}, d_{i,j}^{seg,lane}, k_{i,j}^{seg} \right\}, \tag{3.17}$$

where $id_{i,j}^{seg}$ is a specific identification number. The starting and ending node of a segment are n_1 and n_2, with the coordinates $\left(x_{i,j}^{seg,n1}, y_{i,j}^{seg,n1}\right)$ and $\left(x_{i,j}^{seg,n2}, y_{i,j}^{seg,n2}\right)$ in ENU space. $d_{i,j}^{seg,lane}$ is the width of lanes contained in the segment. Normally, the lane width of straight road is 3.5 meter, while the width of corner is 5 meter. $k_{i,j}^{seg}$ is the slope of the straight line connecting n_1 and n_2. The expression of $k_{i,j}^{seg}$ is:

$$k_{i,j}^{seg} = \left(y_{i,j}^{seg,n2} - y_{i,j}^{seg,n1}\right) / \left(x_{i,j}^{seg,n2} - x_{i,j}^{seg,n1}\right). \tag{3.18}$$

Algorithm 3.3.4 Conversion from way property to segment property

Input: - property of a way:

$$way_i = \left\{ id_i^{way}, \{nod_m\}_{m=1...N_{node}^{way}}, v_i^{way,highway}, v_i^{way,lanes}, v_i^{way,oneway} \right\}$$

Output: $N_{node}^{way} - 1$ segments:

$$seg_{i,j} = \left\{ id_{i,j}^{seg}, x_{i,j}^{seg,n1}, y_{i,j}^{seg,n1}, x_{i,j}^{seg,n2}, y_{i,j}^{seg,n2}, n_{i,j}^{seg,lane}, d_{i,j}^{seg,lane}, k_{i,j}^{seg} \right\}$$

1: **for** $(i = 1; i \leq N_{way}; i++)$ **do**
2: ▶ average derivative cell slope $\bar{k}_i^{way} = 0$ $\bar{k}_i^{way}$ is used to distinguish a straight road stretch and a roundabout
3: **for** $(j = 1; j \leq N_{node}^{way} - 1; j++)$ **do** compute two starting points and slope of a segment
4: ▶ $\left(x_{i,j}^{seg,n1}, y_{i,j}^{seg,n1}\right) = \left(x_j^{nod}, y_j^{nod}\right)$
5: ▶ $\left(x_{i,j}^{seg,n2}, y_{i,j}^{seg,n2}\right) = \left(x_{j+1}^{nod}, y_{j+1}^{nod}\right)$
6: ▶ calculate $k_{i,j}^{seg}$ using Eq. 3.18
7: **end for**
8: **for** $j = 1; j \leq N_{node}^{way} - 1; j++$ **do** ▷ calculate $\bar{k}_i^{way}$
9: ▶ $\bar{k}_i^{way} = \bar{k}_i^{way} + \left(k_{i,j+1}^{seg} - k_{i,j}^{seg}\right)$
10: **end for**
11: ▶ $\bar{k}_i^{way} = \frac{\bar{k}_i^{way}}{N_{node}^{way}-1}$
12: **if** $\left(v_i^{way,highway} == \text{`}truck\_link\text{'} || \text{`}motorway\_link\text{'}\right)$ && $v_i^{way,lanes} == 1$ **then**
13: **if** $\bar{k}_i^{way} < Th_{k_{way}}$ **then**
14: ▶ current lane width $d_{i,j}^{seg,lane}$ is 5 meter ▷ on/off ramp corner
15: ▶ lane number $n_{i,j}^{seg,lane}$ is 1
16: **else**
17: ▶ current lane width $d_{i,j}^{seg,lane}$ is 3.5 meter ▷ on/off ramp straight lane
18: ▶ lane number $n_{i,j}^{seg,lane}$ is 1
19: **end if**
20: **else**
21: **if** $v_i^{way,oneway} == \text{`}yes\text{'}$ **then**
22: ▶ current lane width $d_{i,j}^{seg,lane}$ is 3.5 meter ▷ one-way road
23: ▶ lane number $n_{i,j}^{seg,lane} = v_i^{way,lanes}$
24: **else**
25: ▶ current lane width $d_{i,j}^{seg,lane}$ is 3.5 meter ▷ two-way road
26: ▶ lane number $n_{i,j}^{seg,lane} = \frac{v_i^{way,lanes}}{2}$
27: **end if**
28: **end if**
29: **end for**

The segment elements are computed according to way property. $\left(x_{i,j}^{seg,n1}, y_{i,j}^{seg,n1}\right)$ and $\left(x_{i,j}^{seg,n2}, y_{i,j}^{seg,n2}\right)$ are the coordinates of nod_j and nod_{j+1} in way_i. Then the slope $k_{i,j}^{seg}$ is derived using Eq. 3.18. $n_{i,j}^{seg,lane}$ and $d_{i,j}^{seg,lane}$ are estimated with respect to way properties $v_i^{way,highway}$, $v_i^{way,lanes}$ and $v_i^{way,oneway}$. The segment lane number $n_{i,j}^{seg,lane}$

is determined by the way lane number $v_i^{way,lanes}$ and the oneway sign $v_i^{way,oneway}$. If $v_i^{way,oneway} = Yes$, $n_{i,j}^{seg,lane} = v_i^{way,lanes}$; if $v_i^{way,oneway} = No$, the lane number includes both sides, $n_{i,j}^{seg,lane} = v_i^{way,lanes}/2$. The lane width $d_{i,j}^{seg,lane}$ is derived by $v_i^{way,highway}$ and the segment slopes in a way $k_{i,j}^{seg} \quad (j = 1, ..., N_{node}^{way})$. Normally, a wider lane width appears only at the corners, for example, at some highway on-ramps. Another cue is that at corner road, the segment slopes vary strongly. The conversion procedure from way to segment is illustrated in Algorithm 3.3.4.

And Table 3.1 enumerates all the specific conditions met in the transmission from way to segment. Specifically, the width of link road (e.g. trunk link road and motorway link road in Table 3.1) is unable to determine by only the OSM map. Because the straight link roads own a width of 3.5 meter, while the width of roundabout link roads are 5 meter. Therefore, a strategy to judge a relative straight link road or a roundabout is required. To this end, an average derivative cell slope $\bar{k}_i^{way}$of the i^{th} way is defined, which represents the slope variation degree of a way. A threshold $Th_{k_{way}}$ is set to differentiate these two link roads, and thus the road width is determined.

Table 3.1: *Relationship from OSM road type to lane number and lane width.*

"node-way" topology			"marking" topology	
highway value $v_i^{way,highway}$	lanes value $v_i^{way,lanes}$	oneway value $v_i^{way,oneway}$	lane number $n_{i,j}^{seg,lane}$	lane width $d_{i,j}^{seg,lane}$
trunk_link	1	yes	1	3.5/5 meter
motorway_link	1	yes	1	3.5/5 meter
motorway_link	2	yes	2	3.5 meter
motorway	3	yes	3	3.5 meter
motorway	4	yes	4	3.5 meter
trunk	1	yes	1	3.5 meter
trunk	2	yes	2	3.5 meter
trunk	3	yes	3	3.5 meter
primary	2	no/none	1	3.5 meter
primary	2	yes	2	3.5 meter

3.3.3 Cell

The last unit for lane marking map construction is a "cell". A segment contains one or more road lanes in real road environments, each lane is a "cell". For instance, in Fig. 3.4, the road owns two lanes, so two cells are included in a segment. The m^{th} cell in j^{th} segment is defined as:

$$cel_{i,j,m} = \left\{ id_{i,j,m}^{cel}, x_{i,j,m}^{cel,l1}, y_{i,j,m}^{cel,l1}, x_{i,j,m}^{cel,l2}, y_{i,j,m}^{cel,l2}, x_{i,j,m}^{cel,r1}, y_{i,j,m}^{cel,r1}, x_{i,j,m}^{cel,r2}, y_{i,j,m}^{cel,r2}, b_{i,j,m}^{cel,l}, b_{i,j,m}^{cel,r}, type_{i,j,m}^{cel} \right\}. \tag{3.19}$$

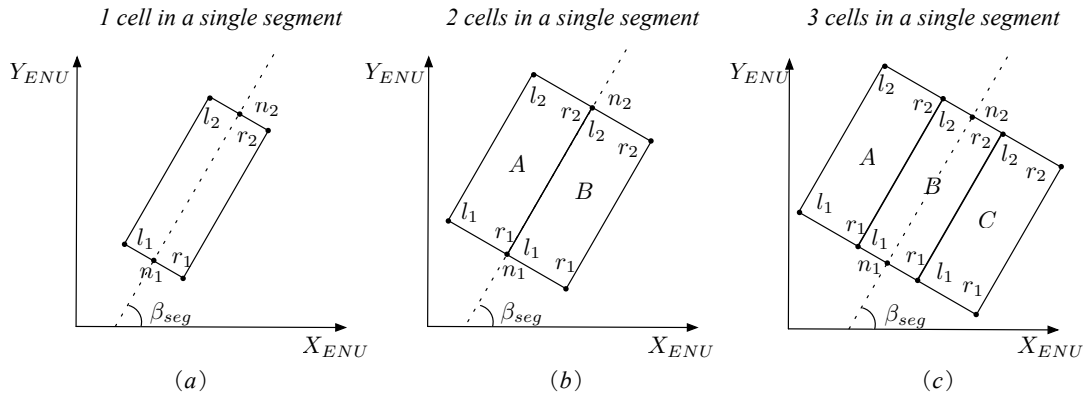


Figure 3.3: *Map reconfiguration. The geometrical relationship between a segment and its attached cells*

In Eq. 3.19, $id^{cel}_{i,j,m}$ is the id number of the m^{th} cell. l_1, l_2, r_1 and r_2 are four vertices of a rectangular cell, as depicted in Fig. 3.2. The coordinate of these vertices in ENU space are $\left(x^{cel,l1}_{i,j,m}, y^{cel,l1}_{i,j,m}\right)$, $\left(x^{cel,l2}_{i,j,m}, y^{cel,l2}_{i,j,m}\right)$, $\left(x^{cel,r1}_{i,j,m}, y^{cel,r1}_{i,j,m}\right)$ and $\left(x^{cel,r2}_{i,j,m}, y^{cel,r2}_{i,j,m}\right)$ respectively. $y = k^{seg}_{i,j} \cdot x + b^{cel,l}_{i,j,m}$ and $y = k^{seg}_{i,j} \cdot x + b^{cel,r}_{i,j,m}$ are the line model of left and right boundaries of the current cell. Cell type $type^{cel}_{i,j,m}$ is assigned as Table 3.2, which is used to represent the m^{th} cell's position in j^{th} segment. For instance, if the host cell is the middle cell in a 3-lane segment, $type^{cel}_{i,j,m} = ThreeLane\_Mid$.

Fig. 3.3 shows the geometrical relationship between a segment and its attached cells. The cell states are determined according to their geometrical relationship. Three cases are considered when a segment transforms to cells: one lane in a segment, two lanes in a segment, and 3 lanes in a segment.

As Fig. 3.3a, only one cell is contained in the segment, which means that the cell and segment overlap. So the cell properties are derived as:

$$
\begin{aligned}
\left(x^{cel,l1}_{i,j,A}, y^{cel,l1}_{i,j,A}\right) &= \left(x^{seg,n1}_{i,j} - \frac{d^{seg,lane}_{i,j}}{2} \cdot sin\beta_{seg}, y^{seg,n1}_{i,j} + \frac{d^{seg,lane}_{i,j}}{2} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,r1}_{i,j,A}, y^{cel,r1}_{i,j,A}\right) &= \left(x^{seg,n1}_{i,j} + \frac{d^{seg,lane}_{i,j}}{2} \cdot sin\beta_{seg}, y^{seg,n1}_{i,j} - \frac{d^{seg,lane}_{i,j}}{2} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,l2}_{i,j,A}, y^{cel,l2}_{i,j,A}\right) &= \left(x^{seg,n2}_{i,j} - \frac{d^{seg,lane}_{i,j}}{2} \cdot sin\beta_{seg}, y^{seg,n2}_{i,j} + \frac{d^{seg,lane}_{i,j}}{2} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,r2}_{i,j,A}, y^{cel,r2}_{i,j,A}\right) &= \left(x^{seg,n2}_{i,j} + \frac{d^{seg,lane}_{i,j}}{2} \cdot sin\beta_{seg}, y^{seg,n2}_{i,j} - \frac{d^{seg,lane}_{i,j}}{2} \cdot cos\beta_{seg}\right), \\
type^{cel}_{i,j,m} &= OneLane,
\end{aligned}
\tag{3.20}
$$

where β_{seg} is the bearing of vector ${}^{ENU}\mathbf{n_1 n_2}$.

Table 3.2: *Assignment of cell type* $type^{cel}_{i,j,m}$.

$type^{cel}_{i,j,m}$ value	Meaning
$OneLane$	The only cell in current segment
$TwoLane\_Left$	The left cell in two-lane segment
$TwoLane\_Right$	The right cell in two-lane segment
$ThreeLane\_Left$	The left cell in three-lane segment
$ThreeLane\_Mid$	The middle cell in three-lane segment
$ThreeLane\_Right$	The right cell in three-lane segment

Fig. 3.3b shows the situation of two "cells" in one "segment", the two cell states are computed as Eq. 3.21 and Eq. 3.22.

In cell A,

$$\begin{aligned}
\left(x^{cel,l1}_{i,j,A}, y^{cel,l1}_{i,j,A}\right) &= \left(x^{seg,n1}_{i,j} - d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n1}_{i,j} + d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,r1}_{i,j,A}, y^{cel,r1}_{i,j,A}\right) &= \left(x^{seg,n1}_{i,j}, y^{seg,n1}_{i,j}\right), \\
\left(x^{cel,l2}_{i,j,A}, y^{cel,l2}_{i,j,A}\right) &= \left(x^{seg,n2}_{i,j} - d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n2}_{i,j} + d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,r2}_{i,j,A}, y^{cel,r2}_{i,j,A}\right) &= \left(x^{seg,n2}_{i,j}, y^{seg,n2}_{i,j}\right), \\
type^{cel}_{i,j,m} &= TwoLane_Left.
\end{aligned} \tag{3.21}$$

In cell B,

$$\begin{aligned}
\left(x^{cel,l1}_{i,j,B}, y^{cel,l1}_{i,j,B}\right) &= \left(x^{seg,n1}_{i,j}, y^{seg,n1}_{i,j}\right), \\
\left(x^{cel,r1}_{i,j,B}, y^{cel,r1}_{i,j,B}\right) &= \left(x^{seg,n1}_{i,j} + d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n1}_{i,j} - d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,l2}_{i,j,B}, y^{cel,l2}_{i,j,B}\right) &= \left(x^{seg,n2}_{i,j}, y^{seg,n2}_{i,j}\right), \\
\left(x^{cel,r2}_{i,j,B}, y^{cel,r2}_{i,j,B}\right) &= \left(x^{seg,n2}_{i,j} + d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n2}_{i,j} - d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
type^{cel}_{i,j,m} &= TwoLane_Right.
\end{aligned} \tag{3.22}$$

As Fig. 3.3c, three cells $cel_{i,j,A}$, $cel_{i,j,B}$ and $cel_{i,j,C}$ are included in current segment $seg_{i,j}$. The properties of the three cells are given as Eq. 3.23 to Eq. 3.25.

In cell A,

$$\begin{aligned}
\left(x^{cel,l1}_{i,j,A}, y^{cel,l1}_{i,j,A}\right) &= \left(x^{seg,n1}_{i,j} - \frac{3}{2}d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n1}_{i,j} + \frac{3}{2}d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,lr1}_{i,j,A}, y^{cel,r1}_{i,j,A}\right) &= \left(x^{seg,n1}_{i,j} - \frac{1}{2}d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n1}_{i,j} + \frac{1}{2}d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,l2}_{i,j,A}, y^{cel,l2}_{i,j,A}\right) &= \left(x^{seg,n2}_{i,j} - \frac{3}{2}d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n2}_{i,j} + \frac{3}{2}d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
\left(x^{cel,lr2}_{i,j,A}, y^{cel,r2}_{i,j,A}\right) &= \left(x^{seg,n2}_{i,j} - \frac{1}{2}d^{seg,lane}_{i,j} \cdot sin\beta_{seg}, y^{seg,n2}_{i,j} + \frac{1}{2}d^{seg,lane}_{i,j} \cdot cos\beta_{seg}\right), \\
type^{cel}_{i,j,m} &= ThreeLane_Left.
\end{aligned} \tag{3.23}$$

In cell B,

$$\begin{aligned}
\left(x_{i,j,A}^{cel,l1}, y_{i,j,A}^{cel,l1}\right) &= \left(x_{i,j}^{seg,n1} - \frac{1}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n1} + d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
\left(x_{i,j,B}^{cel,r1}, y_{i,j,B}^{cel,r1}\right) &= \left(x_{i,j}^{seg,n1} + \frac{1}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n1} - \frac{1}{2}d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
\left(x_{i,j,A}^{cel,l2}, y_{i,j,A}^{cel,l2}\right) &= \left(x_{i,j}^{seg,n2} - \frac{1}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n2} + d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
\left(x_{i,j,B}^{cel,r2}, y_{i,j,B}^{cel,r2}\right) &= \left(x_{i,j}^{seg,n2} + \frac{1}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n2} - \frac{1}{2}d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
type_{i,j,m}^{cel} &= ThreeLane_Mid.
\end{aligned} \tag{3.24}$$

In cell C,

$$\begin{aligned}
\left(x_{i,j,C}^{cel,l1}, y_{i,j,C}^{cel,l1}\right) &= \left(x_{i,j}^{seg,n1} + \frac{1}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n1} - \frac{1}{2}d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
\left(x_{i,j,C}^{cel,r1}, y_{i,j,C}^{cel,r1}\right) &= \left(x_{i,j}^{seg,n1} + \frac{3}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n1} - \frac{3}{2}d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
\left(x_{i,j,C}^{cel,l2}, y_{i,j,C}^{cel,l2}\right) &= \left(x_{i,j}^{seg,n2} + \frac{1}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n2} - \frac{1}{2}d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
\left(x_{i,j,C}^{cel,r2}, y_{i,j,C}^{cel,r2}\right) &= \left(x_{i,j}^{seg,n2} + \frac{3}{2}d_{i,j}^{seg,lane} \cdot sin\beta_{seg}, y_{i,j}^{seg,n2} - \frac{3}{2}d_{i,j}^{seg,lane} \cdot cos\beta_{seg}\right), \\
type_{i,j,m}^{cel} &= ThreeLane_Right.
\end{aligned} \tag{3.25}$$

Hence, the map sources are reconfigured into cells, the left and right lane marking of a cell are represented as the connection of point $\left(x_{i,j,m}^{cel,l1}, y_{i,j,m}^{cel,l1}\right)$ to point $\left(x_{i,j,m}^{cel,l2}, y_{i,j,m}^{cel,l2}\right)$, and point $\left(x_{i,j,m}^{cel,r1}, y_{i,j,m}^{cel,r1}\right)$ to point $\left(x_{i,j,m}^{cel,r2}, y_{i,j,m}^{cel,r2}\right)$ respectively.

Consequently, the map database is mapped to lane markings in ENU space, throughout the transmission from nodes and ways to segments and cells, as shown in Fig. 3.4. In Fig. 3.4, the relationship between the road units and real road BEV image is depicted. The left image contains nodes and ways derived from OSM. This road stretch is a single way with two lines, which are located at the middle of the real road. The right image includes segments and cells, as well as the constructed lane markings from OSM, shown as red curves. The way represented is divided into two segments, one straight part each. And each segment is transformed to two cells, because of the 2-lane road. When all the cells are derived according to the nodes and ways required, a map of lane markings is constructed.

3.4 Lane Marking based Particle Filter

In Chapter 2, the lane marking models are detected using MKE based method. Assume that the rover drives along the road lane, the marking model is regarded as the vehicle driving trajectory during the current time interval. Therefore, marking models are

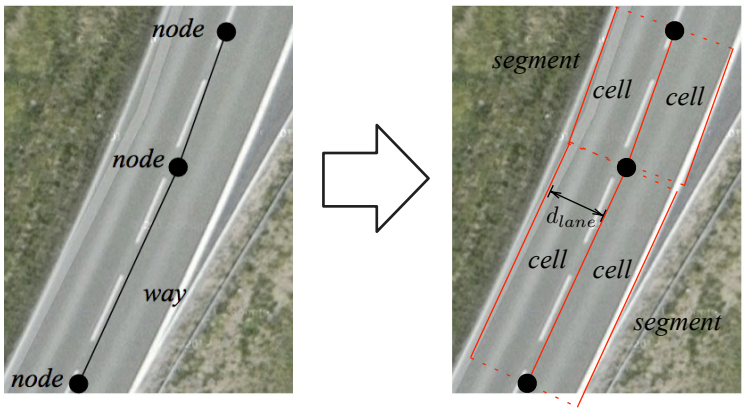


Figure 3.4: *Map reconfiguration to "lane marking" topology.*

exploited as dynamic evolution model in particle filter to assist vehicle localization. A lane marking based particle filter is designed to estimate vehicle positions in ENU space using the marking models derived from vision. This filter can be regarded as a first guess of vehicle localization

A SIR particle filter, introduced in 3.2.3, performs three operations sequentially: prediction, update and resampling. In the following, SIR filter framework is used to adjust rough vehicle position from low-cost GPS receiver, according to the vision-based lane marking models derived in Chapter 2. Particles are created in initialization step at time 0. Estimated particle position is obtained with respect to the dynamic evolution model in prediction step. After that, weights of all particles are computed and normalized according to the latest measurement in update step. In resampling part, SIR algorithm is introduced to deal with degeneracy problem. Finally, the vehicle position is estimated according to the states and weights of all the particles.

Before filter initialization, vehicle forward velocity $v_{veh,k}$ and vehicle yaw angle $\gamma_{veh,k}$ at time k should be estimated according to the low-cost on-vehicle GPS position and the map topology constructed in Section 3.3. To obtain the forward velocity $v_{veh,k}$, the GPS signal at time k, to the GPS position N_v^{pf} time intervals before, are required. Because original GPS signals are noised, $v_{veh,k}$ and $\gamma_{veh,k}$ cannot be estimated directly by connecting $(x_{veh,k}, y_{veh,k})$ and $\left(x_{veh,k-N_v^{pf}}, y_{veh,k-N_v^{pf}}\right)$. Map information is employed to approximate vehicle speed and heading. Assume that the rover is onto road segment $seg_{i,j}$, the state of segment in Eq. 3.17 is required.

Fig. 3.5(a) depicts a simple situation for vehicle and heading estimation. P_1 and P_2 represent the vehicle position at time $k - N_v^{pf}$ and k respectively. P_1' and P_2' are projected points of P_1 and P_2 on road segment $seg_{i,j}$. The distance between P_1' and P_2' are approximated as the motion distance from time $k - N_v^{pf}$ to k, represented as the

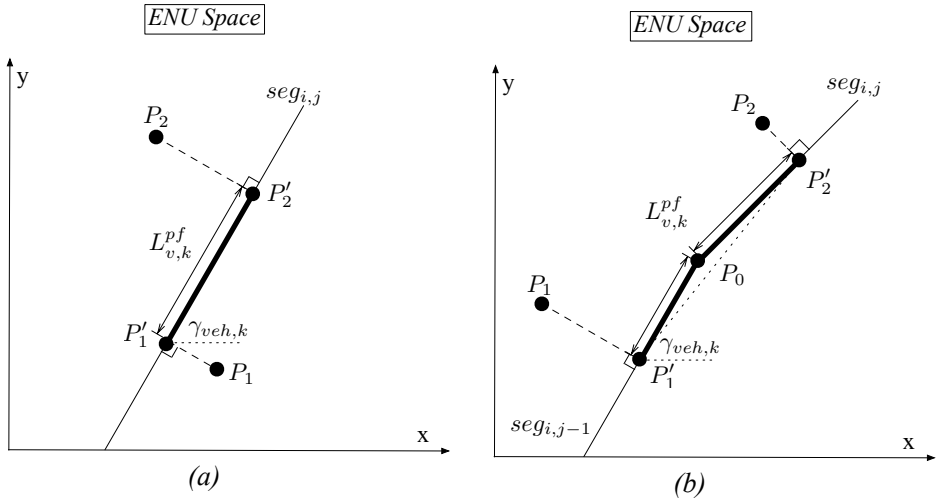


Figure 3.5: *Vehicle speed and yaw angle estimation through GPS positions and map. (a) One segment situation; (b) two segments situation.*

bold part in Fig. 3.5(a). Thus, vehicle speed is estimated as:

$$v_{veh,k} = \frac{L_{v,k}^{pf}}{\left(N_v^{pf} + 1\right) \cdot T_{pf}}, \tag{3.26}$$

where $L_{v,k}^{pf}$ is the distance between P_1' and P_2', T_{pf} is the time interval of a filter cycle. The vehicle yaw angle is valued as$\gamma_{veh,k} = \beta_{seg}$, where β_{seg} is the segment bearing mentioned in Eq. 3.20.

A complicate situation is that P_1 and P_2 are in two neighborhood segments $seg_{i,j-1}$ and $seg_{i,j}$, shown in Fig. 3.5(b). P_1 is nearer to segments $seg_{i,j-1}$ and P_2 is more close to $seg_{i,j}$, so P_1 and P_2 corresponds to $seg_{i,j-1}$ and $seg_{i,j}$ respectively. The procedure to obtain $v_{veh,k}$ and $\gamma_{veh,k}$ are different from simple case. At first, the projection P_1' and P_2' on $seg_{i,j}$ and $seg_{i,j-1}$ are located. The distance $L_{v,k}^{pf}$ is sum length of the polyline between P_1' and P_2', which is the sum of length from P_1' to P_0 and length from P_0 to P_2', represented as the bold parts in Fig. 3.5(b). Then the vehicle speed is achieved using Eq. 3.26. The rover yaw angle $\gamma_{veh,k}$ is the bearing of vector ${}^{ENU}\mathbf{p_1'p_2'}$.For more complex situation, similar methods as two segments procedure are exploited.

The vehicle dynamics, location, forward speed and bearing, are determined from rough GPS position. The vehicle positioning measurement:

$$s_{veh,k} = \left[x_{veh,k}, y_{veh,k}, v_{veh,k}, \gamma_{veh,k}\right]^T, \tag{3.27}$$

is determined completely.

3.4.1 Initialization Step

In this step, N_{pf} particles at time $k = 0$ are created according to vehicle states, where N_{pf} is the total number of particles. The i^{th} particle state at time k is composed of coordinates in ENU space, forward velocity and particle bearing, is represented as:

$$s_k^{(i)} = \left[x_{enu,k}^{(i)}, y_{enu,k}^{(i)}, v_{enu,k}^{(i)}, \gamma_{enu,k}^{(i)}\right]^T, \tag{3.28}$$

where $\left(x_{enu,k}^{(i)}, y_{enu,k}^{(i)}\right)$ are the particle coordinates in meter in ENU space, $v_{enu,k}^{(i)}$ is the forward speed of the particle in m/s and $\gamma_{enu,k}^{(i)}$ is the particle heading in degree.
The initialized locations $x_{enu,0}^{(i)}$ and $y_{enu,0}^{(i)}$ of i^{th} particle are created following uniform distributions $X_{x_{enu,0}}^{pf}$ and $X_{y_{enu,0}}^{pf}$ along both x and y axis centering at the initial vehicle coordinate $(x_{veh,0}, y_{veh,0})$, as follows:

$$X_{x_{enu,0}}^{pf} \sim \mathcal{CN}\left(x_{veh,0}, \left(\sigma_{x_{enu}}^{pf}\right)^2\right), \tag{3.29}$$

$$X_{y_{enu,0}}^{pf} \sim \mathcal{CN}\left(y_{veh,0}, \left(\sigma_{y_{enu}}^{pf}\right)^2\right), \tag{3.30}$$

where $(x_{veh,0}, y_{veh,0})$ is the initial GNSS signal of vehicle in ENU space, $\sigma_{x_{enu}}^{pf}$ and $\sigma_{y_{enu}}^{pf}$ are the standard deviation of distributions $X_{x_{enu,0}}^{pf}$ and $X_{y_{enu,0}}^{pf}$.
The particle velocity is estimated according to vehicle speed in Eq. 3.26. Particle velocity $v_{enu,0}^{(i)}$ is assigned according to a normal distribution $X_{v_{enu,0}}^{pf}$ as Eq. 3.31.

$$X_{v_{enu,0}}^{pf} \sim \mathcal{CN}\left(v_{veh,0}, \left(\sigma_{v_{veh}}^{pf}\right)^2\right), \tag{3.31}$$

where $v_{veh,0}$ is the vehicle velocity at time stamp $k = 0$, $\sigma_{v_{veh}}^{pf}$ is the standard deviation of $X_{v_{veh,0}}^{pf}$.
And the particle yaw angle $\gamma_{enu,0}^{(i)}$ is given following a normal distribution:

$$X_{\gamma_{enu,0}}^{pf} \sim \mathcal{CN}\left(\gamma_{veh,0}, \left(\sigma_{\gamma_{veh}}^{pf}\right)^2\right), \tag{3.32}$$

where $\gamma_{veh,0}$ is the rover yaw angle at time stamp $k = 0$, $\sigma_{\gamma_{veh}}^{pf}$ is the standard deviation of $X_{\gamma_{veh,0}}^{pf}$. Remarkably, since the normal distributed errors are one of the most frequent in measurement, the particle states are estimated with respect to normal distributions Goodman (1963).

At time 0, all the particles have the same importance, so the initialized weight of each particle is valued equally as:

$$\bar{w}_{pf,0}^{(i)} = \frac{1}{N_{pf}}, \quad i = 1, ..., N_{pf}. \tag{3.33}$$

3.4.2 Prediction Step

In this step, particle motion is determined according to dynamic evolution model. To this end, an estimation strategy is designed to accomplish prediction. In previous step, the particles are created with respect to the vehicle dynamics. These particles are depicted as green dots around the vehicle in Fig. 3.6a. The aim is to find the particle positions at the next time, shown as red dots in Fig. 3.6a. The relative translation of a particle is obtained in BEV space, and is projected back to ENU space to determine the particle position at the next stamp.

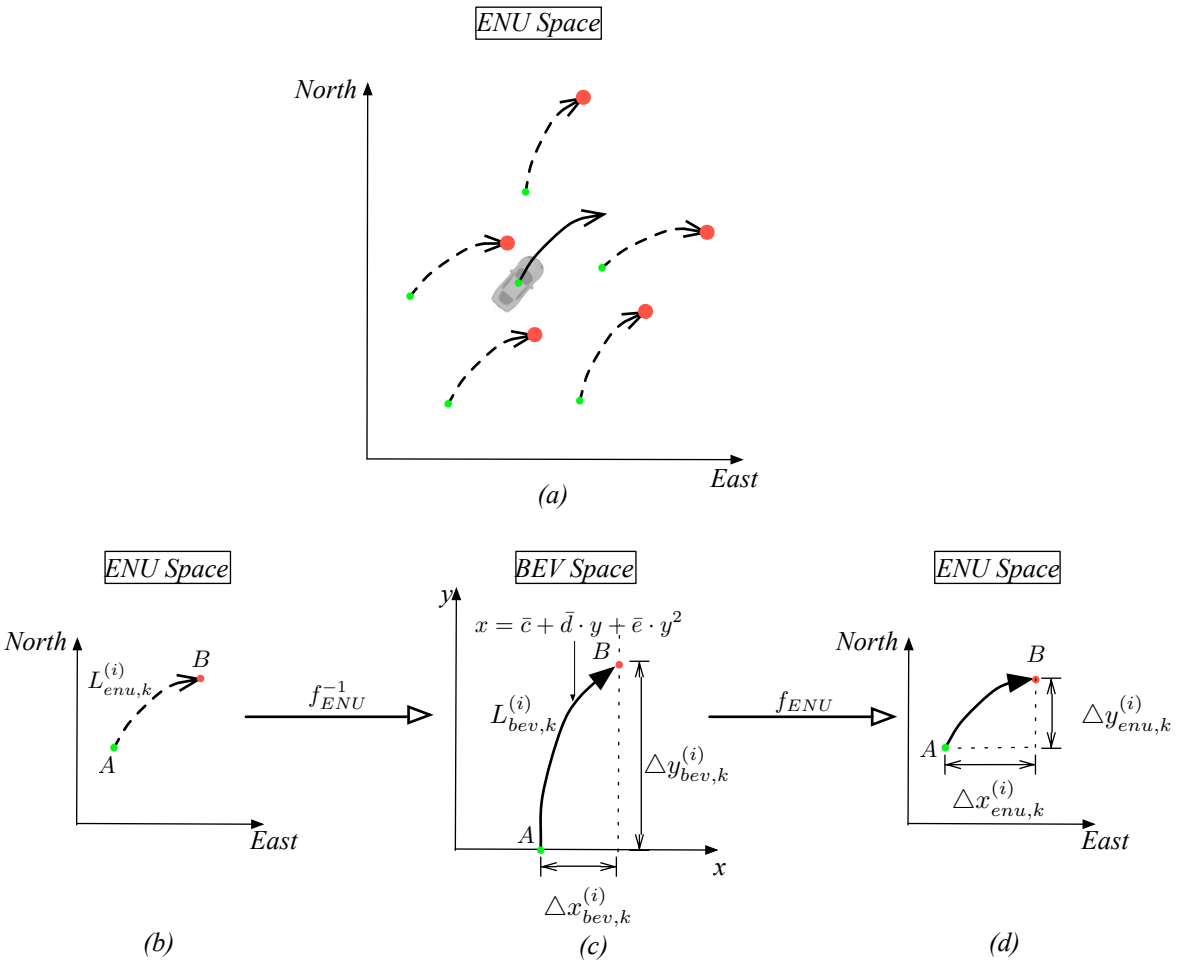


Figure 3.6: *Predicting step of particle filter.*

An example of particle motion from A to B is shown in Fig. 3.6a. At time $k-1$, particle states $s_{k-1}^{(i)}$ are calculated, the goal is to predict the estimated particle position $\left(x_{enu,k}^{(i)}, y_{enu,k}^{(i)}\right)$ using the detected vision-based lane marking models from Eq. 2.22 and Eq. 2.32. The travel distance of each particle in ENU space is represented as:

$$L_{enu,k}^{(i)} = v_{enu,k}^{(i)} \cdot T_{pf}, \tag{3.34}$$

where T_{pf} is the time interval of a filter cycle. Shown in Fig. 3.6b-c, $L_{enu,k}^{(i)}$ is then

projected to BEV space as:

$$L_{bev,k}^{(i)} = \frac{L_{enu,k}^{(i)}}{\lambda_{b2e}}, \tag{3.35}$$

where $\lambda_{b2e} = 0.1$ is a scale factor. The mapping between BEV and ENU spaces is detailed in Appendix A.

In BEV space lane marking is approximated as the rover driving trail. The trajectory length along marking model $x = \bar{c} + \bar{d} \cdot y + \bar{e} \cdot y^2$ from the starting point $(\bar{c}, 0)$, point A in Fig. 3.6c, to a certain point (x, y), point B in Fig. 3.6c, on the curve is expressed as:

$$L_{tra,k}^{(i)}(y_{bev}) = \int_0^{y_{bev}} \sqrt{1 + (\partial x_{bev}/\partial y_{bev})^2} dy_{bev}. \tag{3.36}$$

Introducing the integration expression:

$$f_{int}^{pf}(x) = \int \sqrt{x^2+1} dx = \frac{x}{2}\sqrt{x^2+1} + \frac{1}{2}\ln\left(x + \sqrt{x^2+1}\right) + C, \tag{3.37}$$

and substituting $\bar{c} + \bar{d} \cdot y + \bar{e} \cdot y^2$ for x, Eq. 3.36 is deduced to

$$\begin{aligned} L_{tra,k}^{(i)}(y_{bev}) &= f_{int}^{pf}(y_{bev}) - f_{int}^{pf}(0) \\ &= \frac{1}{2e}\{\frac{\bar{d} + 2\bar{e}y_{bev}}{2}\sqrt{(\bar{d} + 2\bar{e}y_{bev})^2 + 1} + \frac{1}{2}ln[(\bar{d} + 2\bar{e}y_{bev}) \\ &\quad + \sqrt{(\bar{d} + 2\bar{e}y_{bev})^2 + 1}] - \frac{\bar{d}}{2}\sqrt{\bar{d}^2 + 1} - \frac{1}{2}ln[\bar{d} + \sqrt{\bar{d}^2 + 1}]\}, \end{aligned} \tag{3.38}$$

where

$$\bar{c} = \frac{c_l^* + c_r^*}{2}, \tag{3.39}$$

$$\bar{d} = \frac{d_l^* + d_r^*}{2}, \tag{3.40}$$

$$\bar{e} = \frac{e_l^* + e_r^*}{2}. \tag{3.41}$$

Because the derived marking models are quasi-parallel, an "average" model is derived with respect to Eq. 3.39 to Eq. 3.41.

Therefore, the relative displacement of a particle is given by:

$$\triangle y_{bev,k}^{(i)} = \underset{y \in (0, M_{bev})}{\arg\min} \left| L_{bev,k}^{(i)} - L_{tra,k}^{(i)}(y) \right|, \tag{3.42}$$

$$\triangle x_{bev,k}^{(i)} = \bar{c} + \bar{d} \cdot \triangle y_{bev,k}^{(i)} + \bar{e} \cdot (\triangle y_{bev,k}^{(i)})^2. \tag{3.43}$$

Specifically, if the detected lane marking is straight line, which means that $\bar{e} = 0$ and the marking model is $x = \bar{c} + \bar{d} \cdot y$, the definite integral expression in Eq. 3.36 can be calculated directly, so Eq. 3.42 and Eq. 3.43 are simplified to:

$$\triangle y_{bev,k}^{(i)} = \frac{L_{bev,k}^{(i)}}{\sqrt{1 + \left(\bar{d}\right)^2}}, \tag{3.44}$$

$$\triangle x_{bev,k}^{(i)} = \bar{c} + \bar{d} \cdot \triangle y_{bev,k}^{(i)}. \tag{3.45}$$

$\triangle x_{bev,k}^{(i)}$ and $\triangle y_{bev,k}^{(i)}$ are calculated using either parabola model in Eq. 3.42 and Eq. 3.43 or straight line model in Eq. 3.44 and Eq. 3.45. In straight line cases, the time cost is less. As in Fig. 3.6c-d, $\triangle x_{bev,k}^{(i)}$ and $\triangle y_{bev,k}^{(i)}$ are then transformed to ENU space through:

$$\left(\triangle x_{enu,k}^{(i)}, \triangle y_{enu,k}^{(i)}\right) = f_{ENU}\left(\triangle x_{bev,k}^{(i)}, \triangle y_{bev,k}^{(i)}\right). \tag{3.46}$$

f_{ENU} is the mapping from BEV space to ENU space mentioned in Eq. A.13. In conversion function f_{ENU}, particle yaw angle $\gamma_{enu,k-1}^{(i)}$ replaces the heading angle β_{veh} in Eq. A.13. The particle state after movement is then given by:

$$x_{enu,k}^{(i)} = x_{enu,k-1}^{(i)} + \triangle x_{enu,k}^{(i)}, \tag{3.47}$$

$$y_{enu,k}^{(i)} = y_{enu,k-1}^{(i)} + \triangle y_{enu,k}^{(i)}. \tag{3.48}$$

This step is executed every N_{pdt}^{pf} filter time circles, where N_{pdt}^{pf} is the time cycle of prediction.

3.4.3 Update Step

In this step, the weight of each particle $w_{pf,k}^{(i)}$ is updated with respect to the measurement at time k. The importance weight of a particle is calculated according to a four-dimension Gaussian distribution centering at vehicle state $s_{veh,k}$ at time k:

$$w_{pf,k}^{(i)} = \frac{\exp\left(-\frac{\left(x_{enu,k}^{(i)} - x_{veh,k}\right)^2}{2\left(\sigma_{x_{enu}}^{pf}\right)^2} - \frac{\left(y_{enu,k}^{(i)} - y_{veh,k}\right)^2}{2\left(\sigma_{y_{enu}}^{pf}\right)^2} - \frac{\left(v_{enu,k}^{(i)} - v_{veh,k}\right)^2}{2\left(\sigma_{v_{veh}}^{pf}\right)^2} - \frac{\left(\gamma_{enu,k}^{(i)} - \gamma_{veh,k}\right)^2}{2\left(\sigma_{\gamma_{veh}}^{pf}\right)^2}\right)}{4\pi^2 \sigma_{x_{enu}}^{pf} \sigma_{y_{enu}}^{pf} \sigma_{v_{veh}}^{pf} \sigma_{\gamma_{veh}}^{pf}}. \tag{3.49}$$

When $w_{pf,k}^{(i)}$ is derived, normalized weight is computed as:

$$\bar{w}_{pf,k}^{(i)} = w_k^{(i)} / \sum_{i=1}^{N_{pf}} w_{pf,k}^{(i)}. \tag{3.50}$$

3.4.4 Resampling

The particles are resampled according to SIR algorithm. If the weight of a particle $\bar{w}_{pf,k}^{(i)}$ is below a fixed weight threshold $Th_{w_{pf}}$, the particle i is resampled to a certain existing particle j. The probability to become a certain existing particle is proportional to the weight of this existing particle. The state $s_k^{(i)}$ of new propagated particle i is re-updated according to the corresponding particle j, adding normal distribution noises. $x_{enu,k}^{(i)}$ and $y_{enu,k}^{(i)}$ of i^{th} particle are created following uniform distributions $X_{x_{enu,k}}^{pf}$ and $X_{y_{enu,k}}^{pf}$ along both x and y axis centered at the particle j location $\left(x_{enu,k}^{(j)}, y_{enu,k}^{(j)}\right)$, as follows:

$$X_{x_{enu,k}}^{pf} \sim \mathcal{CN}\left(x_{enu,k}^{(j)}, \left(\sigma_{x_{enu}}^{pf}\right)^2\right), \tag{3.51}$$

$$X_{y_{enu,k}}^{pf} \sim \mathcal{CN}\left(y_{enu,k}^{(j)}, \left(\sigma_{y_{enu}}^{pf}\right)^2\right). \tag{3.52}$$

Particle velocity $v_{enu,k}^{(i)}$ is initialized according to a normal distribution $X_{v_{enu,k}}^{pf}$ as Eq. 3.53.

$$X_{v_{enu,k}}^{pf} \sim \mathcal{CN}\left(v_{veh,k}^{(j)}, \left(\sigma_{v_{veh}}^{pf}\right)^2\right), \tag{3.53}$$

where $v_{veh,k}^{(j)}$ is the velocity of particle j. Likewise, the particle yaw angle $\gamma_{enu,k}^{(i)}$ is given following a normal distribution:

$$X_{\gamma_{enu,k}}^{pf} \sim \mathcal{CN}\left(\gamma_{veh,k}^{(j)}, \left(\sigma_{\gamma_{veh}}^{pf}\right)^2\right), \tag{3.54}$$

SIR algorithm is implemented every N_{rsp}^{pf} filtering cycles, where N_{rsp}^{pf} is the time cycle of resampling.

Finally, the approximated vehicle fix is given by:

$$\hat{x}_{veh,k} = \sum_{i=1}^{N_{pf}} \bar{w}_{pf,k}^{(i)} \cdot x_{enu,k}^{(i)}, \tag{3.55}$$

$$\hat{y}_{veh,k} = \sum_{i=1}^{N_{pf}} \bar{w}_{pf,k}^{(i)} \cdot y_{enu,k}^{(i)}. \tag{3.56}$$

The algorithm of the lane marking based particle filter is summarized as Algorithm 3.4.5.

Algorithm 3.4.5 Algorithm of lane marking based particle filter

Input: - input GPS signal (lat, lon, alt)
- lane marking model c_l^*, d_l^*, e_l^*, c_r^*, d_r^*, e_r^*
Output: filtered vehicle position: $(\hat{x}_{veh,k}, \hat{y}_{veh,k})$

▶ conversion (lat, lon, alt) to ENU coordinate $(x_{veh,k}, y_{veh,k})$ according to Appendix A
▶ estimate $s_0^{(i)}$ using Eq. 3.29 to Eq. 3.32
▶ estimate $\bar{w}_{pf,0}^{(i)}$ using Eq. 3.33
while
  do ▷ prediction step
  ▶ estimate $\bar{c}$, $\bar{d}$, $\bar{e}$ using Eq. 3.39 to Eq. 3.41
  for $i = 1; i \leq N_{pf}; i++$ **do**
    ▶ transform $(x_{enu,k}^{(i)}, y_{enu,k}^{(i)})$ to $(x_{bev,k}^{(i)}, y_{bev,k}^{(i)})$ using Eq. A.17
    if $\bar{e} == 0$ **then** ▷ straight line model
      ▶ estimate $\left(\triangle x_{bev,k}^{(i)}, \triangle y_{bev,k}^{(i)}\right)$ from Eq. 3.44 and Eq. 3.45
    else ▷ quadratic model
      ▶ derive $\left(\triangle x_{bev,k}^{(i)}, \triangle y_{bev,k}^{(i)}\right)$ from Eq. 3.42 and Eq. 3.43
    end if
    ▶ transform to $\left(\triangle x_{enu,k}^{(i)}, \triangle y_{enu,k}^{(i)}\right)$ using Eq. 3.46
    ▶ estimate $\left(x_{enu,k}^{(i)}, y_{enu,k}^{(i)}\right)$ from Eq. 3.47 and Eq. 3.48
  end for
end while
if $k\%N_{upd}^{pf} == 0$ **then** ▷ update step
  ▶ estimate vehicle velocity and yaw angle at time k using Eq. 3.26.
  for $i = 1; i \leq N_{pf}; i++$ **do**
    ▶ compute $w_{pf,k}^{(i)}$ using Eq. 3.49
  end for
  for $i = 1; i \leq N_{pf}; i++$ **do**
    ▶ update $\bar{w}_{pf,k}^{(i)}$ using Eq. 3.50
  end for
end if
if $k\%N_{rsp}^{pf} == 0$ **then** ▷ resampling step
  for $i = 1; i \leq N_{pf}; i++$ **do**
    if $\bar{w}_{pf,k}^{(i)} < Th_{w_{pf}}$ **then**
      ▶ resample the i^{th} particle state according to Eq. 3.51 to Eq. 3.54
    end if
  end for
end if
▶ derive $(\hat{x}_{enu,k}, \hat{y}_{enu,k})$ according to Eq. 3.55 and Eq. 3.56

3.5 Analysis on Marking based Particle Filter

In this section, the performance of the proposed particle filter is analyzed, its advantages and drawbacks are enlisted.

The input parameters our filter requires are concise, rough GPS positions as the measurement, and the lane marking model as dynamic evolution model in prediction step. A low-cost GPS receiver and a monocular camera, are easy to implement on real vehicles. This property extends the scope of application on IVs.

Our particle filter employed the detected markings to stimulate vehicle trajectory, so that the estimated predictions are more precise. Normally, the vehicle driving trail is approximated as a straight line in a short period. The rover velocity includes a forward speed and a lateral speed, so the trajectory is more like a parabola than a straight line. In Chapter 2, a parabola is used as the shape model of the lane markings. Assume that the rover moves along the current lane, lane marking models are regarded as the vehicle route in a period.

Additionally, SIR algorithm is exploited in the proposed method to avoid particle degeneracy problem, which is a common problem in particle filter. The weight based particle resampling allows that the low weight particles have opportunities to propagate new particles, which increases the particle diversity.

The parameter tuning is one problem in this method. For instance, the amount of particles N_{pf} affects the time cost and accuracy. Too many particles bring numerous computation, while too few particles decrease the prediction precision. The variances of particle state parameters $\sigma^{pf}_{x_{enu}}$, $\sigma^{pf}_{y_{enu}}$, $\sigma^{pf}_{v_{veh}}$, and $\sigma^{pf}_{\gamma_{veh}}$ are also critical. If the variance values are too low, the dynamic diversity is impacted. If the values are too high, a lot of meaningless particles are created. An improper assignment of weight threshold $Th_{w_{pf}}$ in resampling step also deteriorates the filter performance.

In our particle filter, the vehicle dynamics, speed $v_{veh,k}$ and yaw angle $\gamma_{veh,k}$, are estimated through rough GPS receivers. So $v_{veh,k}$ and $\gamma_{veh,k}$ are not as accurate as the actual vehicle velocity and yaw angle. The particle filter is thus called a first step of vehicle localization. In this marking based particle filter, only the marking shape is employed to predict particle motion. To take full advantage of lane marking information, we can fuse marking locations and map topology together to further improve localization ability. To this end, map information is mentioned as a novel information source in vehicle localization process.

3.6 Map based Lane Selection

The lane markings need to be selected in the constructed lane marking topology. It is possible that lane selection algorithm based on multi-criterion is implemented. At first, the current segment is estimated according to a criterion of the linear distance in ENU space between the rover and segment candidates. Then the current cell is determined by several criteria derived from vision based lane detection and filtered GPS positions. Considering that the left and right boundaries of current cell are exactly the lane markings of current lane, thus when the current cell is determined, the map-based lane markings are derived.

The aim of lane selection is to estimate the current "cell" of the rover in a multi-lane "segment". Lane selection stage is implemented to determine the current cell the vehicle is in two steps: segment selection and cell selection.

3.6.1 Segment Selection

In this step, the map information and filtered vehicle positions are employed to determine the current "segment". In ENU space, let $d_{i,j}^{vs}$ as the linear distance between rover V_{host} and a segment $seg_{i,j}$, shown in Fig. 3.7. The assignment of $d_{i,j}^{vs}$ depends on which zone the rover is in. The segment body of $seg_{i,j}$ (j^{th} segment in i^{th} way) is the gray area in Fig. 3.7. In zone $1, 3, 7, 9$, $d_{i,j}^{vs}$ is the distance of vehicle center and a segment vertex, in zone $2, 4, 6, 8$, $d_{i,j}^{vs}$ is the distance between vehicle center and a segment boundary line, in zone 5, the vehicle is in the segment, $d_{i,j}^{vs} = 0$. The current segment $seg_{i_{cur},j_{cur}}$ is selected via the nearest distance criterion as:

$$(i_{cur}, j_{cur}) = \arg\min_{i,j} d_{i,j}^{vs}.$$

The distance $d_{i,j}^{vs}$ depends number of lanes $n_{i,j}^{seg,lane}$, as well as the zone number where the rover is. Three different cases are considered in the following.

The first case is one-lane segment, which is $n_{i,j}^{seg,lane} = 1$, only one cell $cel_{i,j,A}$ is contained in $seg_{i,j}$. In zone $1, 3, 7, 9$, $d_{i,j}^{vs}$ is the distance of rover and a cell vertex, which is l_1, l_2, r_1 or r_2 of $cel_{i,j,A}$. In zone 2 or 8, $d_{i,j}^{vs}$ is the distance between vehicle center and the top boundary line $y = -\frac{1}{k_{i,j}^{seg}}\left(x - x_{i,j,A}^{cel,l2}\right) + y_{i,j,A}^{cel,l2}$ or bottom boundary line $y = -\frac{1}{k_{i,j}^{seg}}\left(x - x_{i,j,A}^{cel,l1}\right) + y_{i,j,A}^{cel,l1}$ of $seg_{i,j}$. In zone 4 or 6, $d_{i,j}^{vs}$ represents the distance between left line $y = -k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,l}$ or right line $y = -k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,r}$ of $cel_{i,j,A}$. In zone 5, the vehicle is in the segment, so $d_{i,j}^{vs} = 0$.

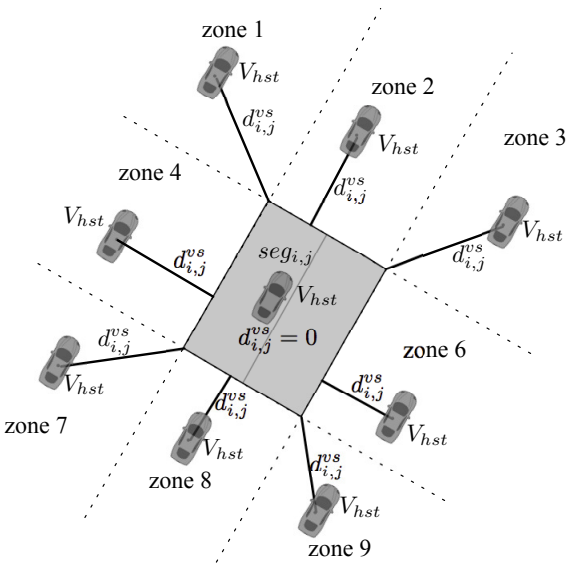


Figure 3.7: *Segment selection.*

$$d^{vs}_{i,j} = \begin{cases} \sqrt{\left(x_{V_{hst}} - x^{cel,l2}_{i,j,A}\right)^2 + \left(y_{V_{hst}} - y^{cel,l2}_{i,j,A}\right)^2}, & (zone \quad 1) \\ \frac{\left|x_{V_{hst}} + k^{seg}_{i,j} \cdot y_{V_{hst}} + k^{seg}_{i,j} \cdot y^{seg,n2}_{i,j} - x^{seg,n2}_{i,j}\right|}{\sqrt{\left(k^{seg}_{i,j}\right)^2 + 1}}, & (zone \quad 2) \\ \sqrt{\left(x_{V_{hst}} - x^{cel,r2}_{i,j,A}\right)^2 + \left(y_{V_{hst}} - y^{cel,r2}_{i,j,A}\right)^2}, & (zone \quad 3) \\ \frac{\left|k^{seg}_{i,j} \cdot x_{V_{hst}} - y_{V_{hst}} + b^{cel,l}_{i,j,A}\right|}{\sqrt{\left(k^{seg}_{i,j}\right)^2 + 1}}, & (zone \quad 4) \\ 0, & (zone \quad 5) \\ \frac{\left|k^{seg}_{i,j} \cdot x_{V_{hst}} - y_{V_{hst}} + b^{cel,r}_{i,j,A}\right|}{\sqrt{\left(k^{seg}_{i,j}\right)^2 + 1}}, & (zone \quad 6) \\ \sqrt{\left(x_{V_{hst}} - x^{cel,l1}_{i,j,A}\right)^2 + \left(y_{V_{hst}} - y^{cel,l1}_{i,j,A}\right)^2}, & (zone \quad 7) \\ \frac{\left|x_{V_{hst}} + k^{seg}_{i,j} \cdot y_{V_{hst}} + k^{seg}_{i,j} \cdot y^{seg,n1}_{i,j} - x^{seg,n1}_{i,j}\right|}{\sqrt{\left(k^{seg}_{i,j}\right)^2 + 1}}, & (zone \quad 8) \\ \sqrt{\left(x_{V_{hst}} - x^{cel,r1}_{i,j,A}\right)^2 + \left(y_{V_{hst}} - y^{cel,r1}_{i,j,A}\right)^2}. & (zone \quad 9) \end{cases} \tag{3.57}$$

The second case is a two-lane segment, which is $n^{seg,lane}_{i,j} = 2$, $cel_{i,j,A}$ and $cel_{i,j,B}$ is contained in $seg_{i,j}$. In zone 1 or 7, $d^{vs}_{i,j}$ is the distance of rover and cell vertex l_1 or l_2 of $cel_{i,j,A}$. In zone 3 or 9, $d^{vs}_{i,j}$ is the distance of rover and cell vertex r_1 or r_2 of the right side cell $cel_{i,j,B}$ in segment $seg_{i,j}$. In zone 2 or 8, $d^{vs}_{i,j}$ is the distance between vehicle center and the top boundary line $y = -\frac{1}{k^{seg}_{i,j}}\left(x - x^{seg,n2}_{i,j}\right) + y^{seg,n2}_{i,j}$ or bottom boundary line $y = -\frac{1}{k^{seg}_{i,j}}\left(x - x^{seg,n1}_{i,j}\right) + y^{seg,n1}_{i,j}$ of $seg_{i,j}$. In zone 4 or 6, $d^{vs}_{i,j}$ represents the distance between left line $y = -k^{seg}_{i,j} \cdot x + b^{cel,l}_{i,j,A}$ of the left cell $cel_{i,j,A}$ or right line $y = -k^{seg}_{i,j} \cdot x + b^{cel,r}_{i,j,B}$ of the right cell $cel_{i,j,B}$. In zone 5, the vehicle is in the segment, so $d^{vs}_{i,j} = 0$.

$$d_{i,j}^{vs} = \begin{cases} \sqrt{\left(x_{V_{hst}} - x_{i,j,A}^{cel,l2}\right)^2 + \left(y_{V_{hst}} - y_{i,j,A}^{cel,l2}\right)^2}, & (zone \quad 1) \\ \frac{\left|x_{V_{hst}} + k_{i,j}^{seg} \cdot y_{V_{hst}} + k_{i,j}^{seg} \cdot y_{i,j}^{seg,n2} - x_{i,j}^{seg,n2}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 2) \\ \sqrt{\left(x_{V_{hst}} - x_{i,j,B}^{cel,r2}\right)^2 + \left(y_{V_{hst}} - y_{i,j,B}^{cel,r2}\right)^2}, & (zone \quad 3) \\ \frac{\left|k_{i,j}^{seg} \cdot x_{V_{hst}} - y_{V_{hst}} + b_{i,j,A}^{cel,l}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 4) \\ 0, & (zone \quad 5) \\ \frac{\left|k_{i,j}^{seg} \cdot x_{V_{hst}} - y_{V_{hst}} + b_{i,j,B}^{cel,r}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 6) \\ \sqrt{\left(x_{V_{hst}} - x_{i,j,A}^{cel,l1}\right)^2 + \left(y_{V_{hst}} - y_{i,j,A}^{cel,l1}\right)^2}, & (zone \quad 7) \\ \frac{\left|x_{V_{hst}} + k_{i,j}^{seg} \cdot y_{V_{hst}} + k_{i,j}^{seg} \cdot y_{i,j}^{seg,n1} - x_{i,j}^{seg,n1}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 8) \\ \sqrt{\left(x_{V_{hst}} - x_{i,j,B}^{cel,r1}\right)^2 + \left(y_{V_{hst}} - y_{i,j,B}^{cel,r1}\right)^2}. & (zone \quad 9) \end{cases} \tag{3.58}$$

The third case is a three-lane segment situation, which means $n_{i,j}^{seg,lane} = 3$, $cel_{i,j,A}$ and $cel_{i,j,B}$ and $cel_{i,j,C}$. In zone 1 or 7, $d_{i,j}^{vs}$ is the distance of rover and cell vertex l_1 or l_2 of $cel_{i,j,A}$. In zone 3 or 9, $d_{i,j}^{vs}$ is the distance of rover and cell vertex r_1 or r_2 of the right side cell $cel_{i,j,C}$ in segment $seg_{i,j}$. In zone 2 or 8, $d_{i,j}^{vs}$ is the distance between vehicle center and the top boundary line $y = -\frac{1}{k_{i,j}^{seg}}\left(x - x_{i,j}^{seg,n2}\right) + y_{i,j}^{seg,n2}$ or bottom boundary line $y = -\frac{1}{k_{i,j}^{seg}}\left(x - x_{i,j}^{seg,n1}\right) + y_{i,j}^{seg,n1}$ of $seg_{i,j}$. In zone 4 or 6, $d_{i,j}^{vs}$ represents the distance between left line $y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,l}$ of the left cell $cel_{i,j,A}$ or right line $y = k_{i,j}^{seg} \cdot x + b_{i,j,C}^{cel,r}$ of the right cell $cel_{i,j,C}$. In zone 5, the vehicle is in the segment, so $d_{i,j}^{vs} = 0$.

$$d_{i,j}^{vs} = \begin{cases} \sqrt{\left(x_{V_{hst}} - x_{i,j,A}^{cel,l2}\right)^2 + \left(y_{V_{hst}} - y_{i,j,A}^{cel,l2}\right)^2}, & (zone \quad 1) \\ \frac{\left|x_{V_{hst}} + k_{i,j}^{seg} \cdot y_{V_{hst}} + k_{i,j}^{seg} \cdot y_{i,j}^{seg,n2} - x_{i,j}^{seg,n2}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 2) \\ \sqrt{\left(x_{V_{hst}} - x_{i,j,C}^{cel,r2}\right)^2 + \left(y_{V_{hst}} - y_{i,j,C}^{cel,r2}\right)^2}, & (zone \quad 3) \\ \frac{\left|k_{i,j}^{seg} \cdot x_{V_{hst}} - y_{V_{hst}} + b_{i,j,A}^{cel,l}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 4) \\ 0, & (zone \quad 5) \\ \frac{\left|k_{i,j}^{seg} \cdot x_{V_{hst}} - y_{V_{hst}} + b_{i,j,C}^{cel,r}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 6) \\ \sqrt{\left(x_{V_{hst}} - x_{i,j,A}^{cel,l1}\right)^2 + \left(y_{V_{hst}} - y_{i,j,A}^{cel,l1}\right)^2}, & (zone \quad 7) \\ \frac{\left|x_{V_{hst}} + k_{i,j}^{seg} \cdot y_{V_{hst}} + k_{i,j}^{seg} \cdot y_{i,j}^{seg,n1} - x_{i,j}^{seg,n1}\right|}{\sqrt{\left(k_{i,j}^{seg}\right)^2 + 1}}, & (zone \quad 8) \\ \sqrt{\left(x_{V_{hst}} - x_{i,j,C}^{cel,r1}\right)^2 + \left(y_{V_{hst}} - y_{i,j,C}^{cel,r1}\right)^2}. & (zone \quad 9) \end{cases} \tag{3.59}$$

3.6.2 Cell Selection

The second step is to determine which cell the vehicle is in, according to multi-criterion. Considered criteria include lane changing criterion, third lane marking criterion, and history vehicle state criterion. Assuming that the maximum lane number is 3, which are left lane, middle lane, and right lane. The corresponding likelihood of each lane are p_l^{ls}, p_m^{ls}, and p_r^{ls}. If it is one-lane situation, only the middle lane p_m^{ls} is considered. If dual-lane, introduce both left and right lanes p_l^{ls}, p_r^{ls}. If 3-lane, consider all p_l^{ls}, p_m^{ls}, and p_r^{ls}.

The first criterion is lane changing, which is determined according to the zero order component c in lane marking model $x = c + d \cdot y + e \cdot y^2$. The strategy is shown as in the left part of Table 3.3, $p_{l,lc}^{ls}$, $p_{m,lc}^{ls}$ and $p_{r,lc}^{ls}$ are the probabilities of left, middle, and right lane respectively. Take one situation as example, if there is a lane changing from a left lane to a right lane in a 3 lane segment, the rover can be considered from the left lane to the middle lane, or from the middle lane to the right lane. So the current lane can be the middle lane or the right lane, which means $p_{l,lc}^{ls} = 0$, $p_{m,lc}^{ls} = 0.5$, $p_{r,lc}^{ls} = 0.5$ respectively.

The second criterion is the third lane marking. At first, third lane marking parameter range is estimated through the detected two current lane markings in Chapter 2. $p_{l,mk}^{ls}$, $p_{m,mk}^{ls}$ and $p_{r,mk}^{ls}$ are the likelihood of third lane marking criteria respectively. According to the third lane detection result, a similar strategy as lane changing criterion is demonstrated as the right part of Table 3.3. For instance, if only the left lane marking is detected in a 3 lane segment, the vehicle can be in the middle lane or in the right lane. But noises such as road barriers and shadows may affect the third marking detection, which leads to some contradictory cases. For instance, a third marking on the left is detected in a one lane segment. In such contradictory cases, the third markings criterion doesn't work, $p_{l,mk}^{ls}$, $p_{m,mk}^{ls}$ and $p_{r,mk}^{ls}$ are valued to 0.

The third criterion is the historical vehicle state, which means the left lane, the middle lane or the right lane the vehicle was in in previous states. The likelihood of historical state are represented as $p_{l,hs}^{ls}$, $p_{m,hs}^{ls}$, and $p_{r,hs}^{ls}$. Therefore, the integrated likelihood of lanes are:

$$p_l^{ls} = k_{lc}^{ls} p_{l,lc}^{ls} + k_{mk}^{ls} p_{l,mk}^{ls} + k_{hs}^{ls} p_{l,hs}^{ls}, \tag{3.60}$$

$$p_m^{ls} = k_{lc}^{ls} p_{m,lc}^{ls} + k_{mk}^{ls} p_{m,mk}^{ls} + k_{hs}^{ls} p_{m,hs}^{ls}, \tag{3.61}$$

$$p_r^{ls} = k_{lc}^{ls} p_{r,lc}^{ls} + k_{mk}^{ls} p_{r,mk}^{ls} + k_{hs}^{ls} p_{r,hs}^{ls}. \tag{3.62}$$

The current lane is selected according to the maximum value of p_l^{ls}, p_m^{ls} and p_r^{ls}.

Table 3.3: *Lane changing criterion and third marking criterion.*

lane changing criterion				
lane number:		1	2	3
left to right	$p_{l,lc}$	0	0	0
	$p_{m,lc}$	0	0	0.5
	$p_{r,lc}$	0	1	0.5
right to left	$p_{l,lc}$	0	1	0.5
	$p_{m,lc}$	0	0	0.5
	$p_{r,lc}$	0	0	0
lane keeping	$p_{l,lc}$	0		
	$p_{m,lc}$	0		
	$p_{r,lc}$	0		

third marking criterion				
lane number:		1	2	3
only left	$p_{l,mk}$	0	0	0
	$p_{m,mk}$	0	0	0.5
	$p_{r,mk}$	0	1	0.5
only right	$p_{l,mk}$	0	1	0.5
	$p_{m,mk}$	0	0	0.5
	$p_{r,mk}$	0	0	0
both	$p_{l,mk}$	0	0	0
	$p_{m,mk}$	0	0	1
	$p_{r,mk}$	0	0	0
none	$p_{l,mk}$	0		
	$p_{m,mk}$	0		
	$p_{r,mk}$	0		

3.7 Marking based Vehicle Localization

The method is to match the two different kinds of lane markings in BEV space, in order to help improve vehicle localization. At first, map based markings in BEV space are detected. Two parts are included, one part includes the left and right lane markings of current cell, and the other owns the lane markings of the corresponding cell in the next segment. Then both the vision based markings and the map based markings are compared and fitted in BEV space, using multi-kernel based estimation. The 2D transformation matrix between these two sources is able to adjust vehicle position filtered from marking based particle filter.

3.7.1 Map based Markings in BEV Space

Map based markings contain not only the markings of current cell, but also the markings of the corresponding cell in the next segment. On one side, the markings of the cell in the next segment ensure adequate marking pixels, even rover comes to the end of a cell. On the other side, due to the length restriction of the sight in BEV space, markings of the corresponding cell two segments forehead is useless. So only the markings of host cell and the corresponding cell in front are selected, the current cell, and the corresponding cell in front.

Let $S_l^{bev} = S_{cur,l}^{bev} \cup S_{nxt,l}^{bev}$ and $S_r^{bev} = S_{cur,r}^{bev} \cup S_{nxt,r}^{bev}$ the sets of left and right marking pixels, $S_{cur,l}^{bev}$ and $S_{cur,r}^{bev}$ mean the marking pixels of current cell, and $S_{nxt,l}^{bev}$ and $S_{nxt,r}^{bev}$ are the marking pixels of the following corresponding cell. All the possible cases are considered to enumerate the relationship between the current cell and the required front cell.

The simplest case is that both the current segment $seg_{i,j}$ and next segment $seg_{i,j+1}$ are

1-lane road sections. So the only cell in segment $seg_{i,j}$ is $cell_{i,j,A}$, while the only cell of segment $seg_{i,j+1}$ is $cell_{i,j+1,A}$. The marking pixels of current cell is expressed as:

$$S_{cur,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,l} \left(x_{i,j,A}^{cell,l1} \leq x \leq x_{i,j,A}^{cell,l2}\right)\right\}, \tag{3.63}$$

$$S_{cur,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,r} \left(x_{i,j,A}^{cell,r1} \leq x \leq x_{i,j,A}^{cell,r2}\right)\right\}. \tag{3.64}$$

The pixel sets $S_{nxt,l}^{bev}$ and $S_{nxt,r}^{bev}$ of the next segment are represented as Eq. 3.65 and Eq. 3.66.

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l} \left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \tag{3.65}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r} \left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \tag{3.66}$$

More complex cases, including the situations where the current and next segments are multi-lane sections, are analyzed in like manner. A complete explanation of $S_{nxt,l}^{bev}$ and $S_{nxt,r}^{bev}$ set up is detailed in Appendix B.

3.7.2 Multi-Kernel based Vehicle Localization

When both vision-based and map-based lane markings are obtained and associated, vehicle positions are optimized using a multi-kernel based estimation method. Fig. 3.8 shows the vehicle localization procedure. On one hand, when a "cell" is selected in the map, the markings of this cell, as well as those of the cell in front are projected to BEV space, as shown the black lines in Fig. 3.8. A pair of lane markings of a rectangle cell are a pair of paralleled lines in map topology, so the map-based markings are combination of several straight lines. The sets of left and right marking pixels are denoted as S_l^{bev} and S_r^{bev} respectively. On the other hand, vision-based lane markings are represented as the form of quadratic model $(c_i^*, d_i^*, e_i^*)\,, i = l, r$ from Eq. 2.22 and Eq. 2.32, as shown the gray curves in Fig. 3.8. The transformation from $(c_i^*, d_i^*, e_i^*)\,, i = l, r$ to S_l^{bev} and S_r^{bev} can be regarded as the movement from rough GNSS data to positions according to map information.

However, it is difficult to compute a transformation matrix from a model (vision-based markings) to a set of pixels (map-based markings). The idea is to at first estimate the transformation from the set of pixels to the marking model. The required matrix is then derived using inverse matrix transform.

A transformation matrix from map-based markings to vision-based lane markings are estimated according to a multi-kernel based estimation method, the same method to

determine marking parameters in Chapter 2. The transformation matrix is defined as $T_{loc}(\triangle x_{loc}, \triangle\theta_{loc})$, where $\triangle x_{loc}$ is lateral displacement, and $\triangle\theta_{loc}$ is vehicle rotation. A multi-kernel based descriptor $G_{pi}(c_i^*, d_i^*, e_i^*, x, y)$, defined in Eq. 2.23, is introduced to describe the likelihood between a single pixel on a map-based marking and a vision-based marking model. The optimized lateral displacement $\triangle x_{loc}^*$ and rotation $\triangle\theta_{loc}^*$ are optimized according to $G_{pi}(c_i^*, d_i^*, e_i^*, x, y)$ as:

$$(\triangle x_{loc}^*, \triangle\theta_{loc}^*) = \underset{\triangle x_{loc}, \triangle\theta_{loc}}{\arg\max} \left[\sum_{i=l,r} \sum_{(x,y)\in S_{i,tra}} G_{pi}(c_i^*, d_i^*, e_i^*, x, y) \right], \tag{3.67}$$

where

$$S_{l,tra} = \{(x,y) \,|\, (x,y,1) = (x',y',1) \cdot T_{loc}, (x',y') \in S_{l,bev}\}, \tag{3.68}$$

$$S_{r,tra} = \{(x,y) \,|\, (x,y,1) = (x',y',1) \cdot T_{loc}, (x',y') \in S_{r,bev}\}. \tag{3.69}$$

Transformed markings through $T_{loc}(\triangle x_{loc}^*, \triangle\theta_{loc}^*)$ are marked as dashed lines in Fig. 3.8. The inverse matrix $T_{loc}^{-1}(\triangle x_{loc}^*, \triangle\theta_{loc}^*)$ of $T_{loc}(\triangle x_{loc}^*, \triangle\theta_{loc}^*)$ is the transformation of lane marking based vehicle localization, from vehicle position filtered by particle filter $(\hat{x}_{bev,k}, \hat{y}_{bev,k})$ to the adjusted vehicle fix $\left(x'_{bev,k}, y'_{bev,k}\right)$. So $\left(x'_{bev,k}, y'_{bev,k}\right)$ is computed as:

$$\left(x'_{bev,k}, y'_{bev,k}, 1\right) = (\hat{x}_{bev,k}, \hat{y}_{bev,k}, 1) \cdot T_{loc}^{-1}(\triangle x_{loc}^*, \triangle\theta_{loc}^*). \tag{3.70}$$

Therefore, the marking based localized position is $\left(x'_{enu,k}, y'_{enu,k}\right)$, translating from BEV space to ENU space.

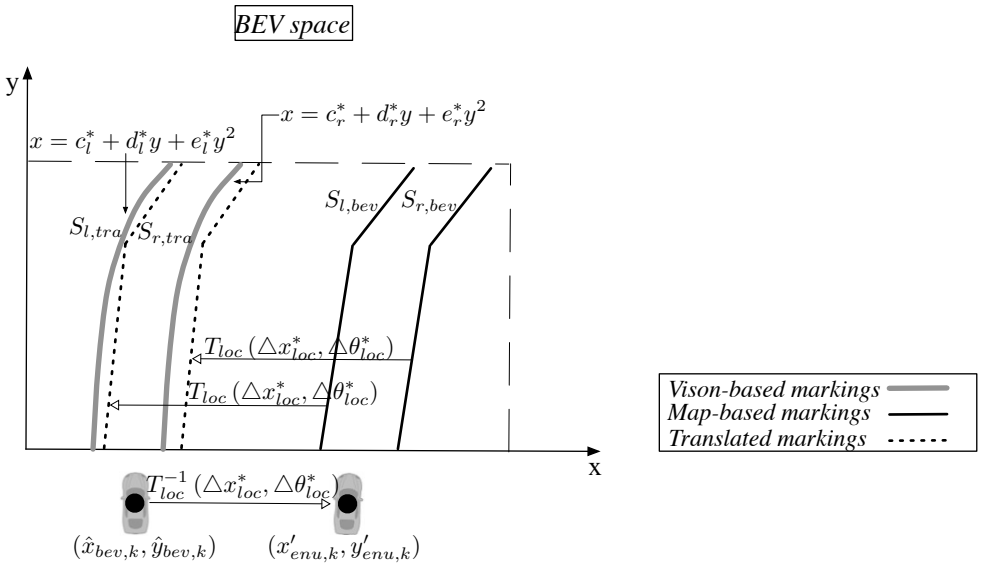


Figure 3.8: *Vehicle localization procedure.*

3.8 Experimental Validation

The proposed method is experimentally validated using the data from an intelligent vehicle platform provided by KITTI Geiger et al. (2013). Since, the considered dataset was acquired using a high-end positioning system, on-road vehicle environment perturbations were modeled by adding uniform distribution noises to the corresponding vehicle fix, speed and yaw angle measurements. A uniform distribution based noise of 10 meters on both horizontal and vertical dimensions Miura and Kamijo (2014) are added on the ground truth vehicle positions, performing as input rough GPS data of our method. The vehicle velocity and yaw angle measurements are noised with uniform distributions of 10 m/s and 5°respectively. The standard deviations in Eq. 3.29 and Eq. 3.30 are assigned as $\sigma^{pf}_{x_{enu}} = 10/3$ and $\sigma^{pf}_{y_{enu}} = 10/3$. The reason of this setting is that the accuracy of normal low cost GPS is 10 meters, and the three standard deviations (3σ) of normal distribution account for 99.7% of the set. To ensure that in prediction step particles can reach the exact location, the variances of 2D coordinate are set as $\sigma^{pf}_{x_{enu}} = 10/3$ and $\sigma^{pf}_{y_{enu}} = 10/3$. In like manner, the standard deviation in Eq. 3.31 is valued as $\sigma^{pf}_{v_{veh}} = 10/3$. In Eq. 3.32 $\sigma^{pf}_{\gamma_{veh}} = 5°/3$ is the standard deviation of $X^{pf}_{\gamma_{veh,0}}$. The assignment of $\sigma^{pf}_{\gamma_{veh}}$ is similar as $\sigma^{pf}_{x_{enu}}$ and $\sigma^{pf}_{y_{enu}}$.

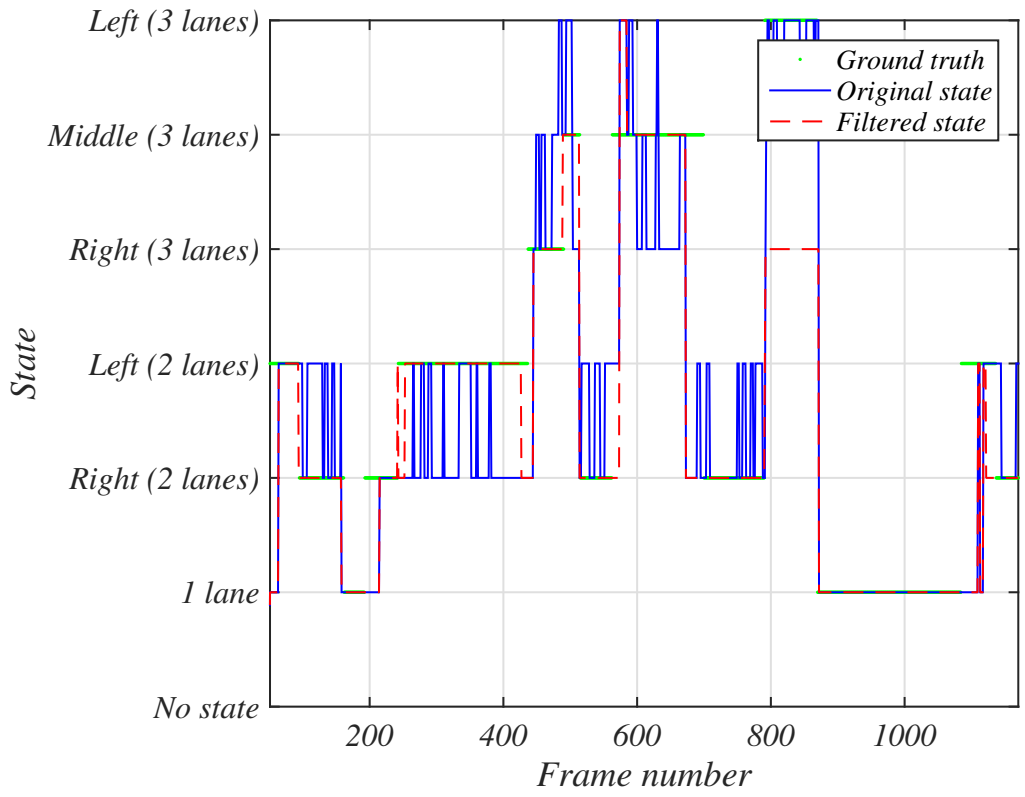


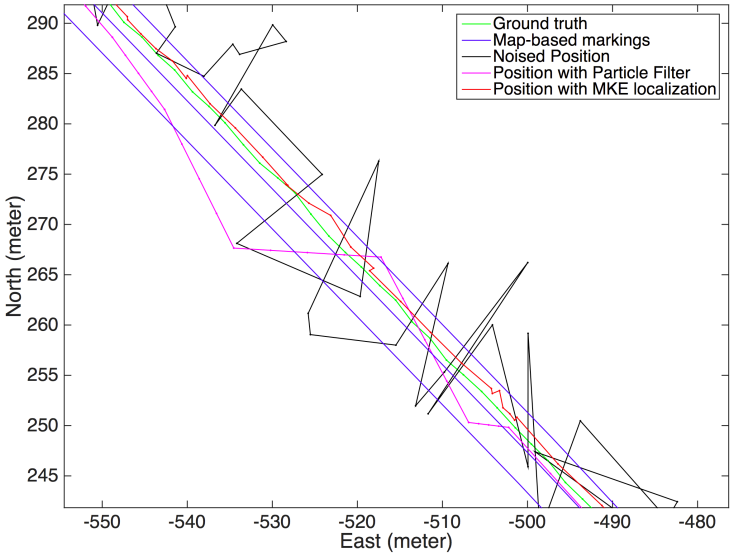
Figure 3.9: *Lane selection result.*

Fig. 3.9 illustrates the result of lane selection. In lane selection part, the coefficients

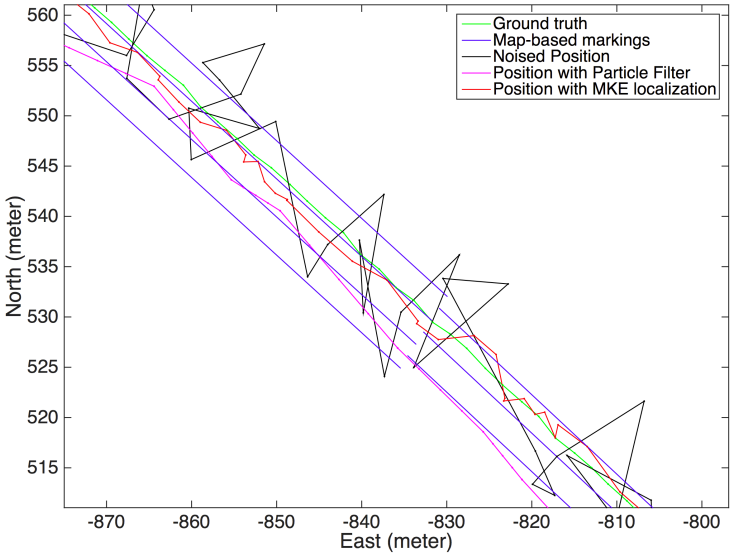
in Eq. 3.60 to Eq. 3.62 are set as $k_{lc} = 0.60$, $k_{mk} = 0.24$ and $k_{hs} = 0.16$, because lane changing behaviors effect lane selection decision most deeply, while the history states play the slightest role to choose the current lane. The ground truth states are labeled manually according to vision images. Original state is the state of the nearest cell to the vehicle, and filtered state is the state estimated from multi-criterion lane selection. The success ratio of original states is 52.42%, while the success ratio of filtered states using lane selection increases to 78.23%. One critical reason of unsuccessful lane selection is that the vehicle position after particle filter is away from the ground truth position along the vehicle forward heading, which leads to a "delay". For instance, at around frame 200, the benchmark position already drives from a one lane segment to a two lane segment, but the filtered vehicle position is still in one lane segment. This "delay" lasts until the filtered vehicle position drives to two lane segment. Another potential reason is noise disturb. At frames around number 800 to 850, road barriers are detected as a third left lane marking by mistake, resulting in a wrong lane selection judgment.

The results of particle filter and marking based localization are depicted in Fig. 3.10. Fig. 3.10a-b are two zoomed map areas. In Fig. 3.10a, the pink curve is the vehicle positions filtered from rough GNSS signals (black), using particle filter. But this pink curve is not in the road cell area which the vehicle is in. The vehicle positions on pink curve are used to select road cell according to multi-criterion. When the road cell is determined, marking based localization is implemented, the vehicle position is adjusted to the central area of road, as red curve in Fig. 3.10a. Fig. 3.10b depicts an exceptional example, a mismatch occurred between two sources, benchmark positions from KITTI and map information from OSM. In this example, the ground truth GPS data is in the middle of two lanes, but in the vision, the vehicle is in the middle lane, obviously, at least one source is not accurate. This mismatch leads to an error even the marking based method is utilized. Fig. 3.10c is a numerical comparison on lateral displacement errors, among noised GPS measurements (cyan), vehicle position after particle filter (pink) and marking-based position (red). In Fig. 3.10c, both the pink and red curves are included in noise error bound. And the red curve is closer to ground truth than the pink curve, which implies that marking based localization improves particle filter based localization.

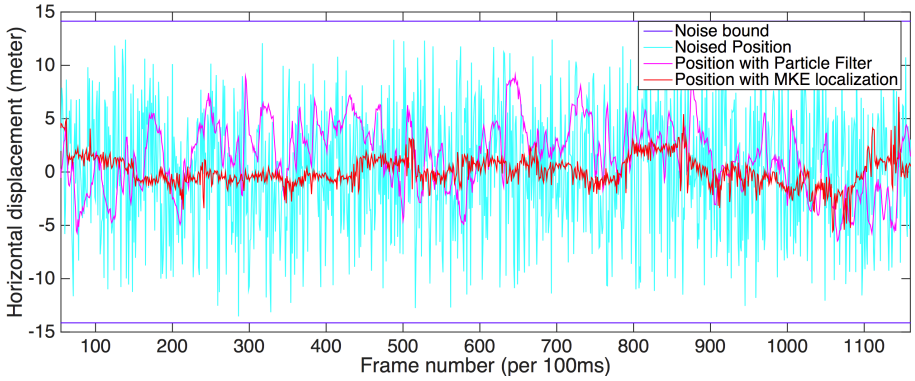
Table 3.4 provides the performance metrics of localization results. In this table, the mean values of both errors are far below the noise error bound (14.2m). The maximum error of particle filter (9.083m) does not exceed the noise error bound neither. Comparing position errors of the two methods, all the statistics of marking based method are less than those of particle filter, which numerically proves that marking based localization helps to improve the performance of vehicle fix.



(a)



(b)



(c)

Figure 3.10: *Localization result. (a)-(b) are zoomed areas in map space; (c) is lateral position error.*

Table 3.4: *Error statistics.*

	Lateral position error after particle filter	Lateral position error after marking based localization
Mean value	1.884m	0.089m
MAE	2.867m	1.006m
Standard deviation	2.942m	1.284m
Max	9.083m	5.429m
95th percentile	6.345m	2.589m

3.9 Conclusion

A lane marking based vehicle localization technique, exploiting rough GPS, speed and yaw angle measurements and an open source map, has been demonstrated and experimentally validated. The results verify a real-time and precise vehicle localization. A "lane marking" map topology is created according to OSM database at first. Then the lane markings of the current lane are selected through a multi-criterion method. The vehicle position is adjusted in two steps. Firstly, a particle filter is designed to adjust the rough vehicle position. Secondly, vision-based markings and map-based markings are fused to enhance vehicle position, using a multi-kernel estimation method.

Chapter 4

Error Modeling

Contents

4.1 Introduction

Chapter 3 proposed the lane marking based vehicle localization using low-cost GPS and OSM. From the camera source, the lane markings signs in front of the rover are detected. From OSM source, a map of lane markings is constructed in ENU space. The detected lane markings, the "marking-topology" map, and the vehicle positioning from low-cost GPS are fused to implement marking based particle filter, multi-cue based lane selection, and multi-kernel based vehicle localization. This procedure improves the localization performance, but we notice that some unusual situations (the MAE of about 7% of the frames are greater than 4 meter) occur in our localization algorithm. In these situations, the positioning errors are not reduced using our localization method, instead, the localized coordinates contain larger error than rough GPS coordinates in some positions.

Analyzing the operation of our approach on real traffic data, we discovered that these unusual localization processes occurred in the vehicle localization procedure can be detected. Most of these errors appear in some specific steps. For instance, an improper lane marking detection produces wrong marking shape, which differs a lot from the "marking topology" map from OSM database, and thus disturbs the localization results. In the multi-kernel estimation step, the derived marking "signatures" from vision and those from OSM are fitted to obtain the transformation matrix for location correction. We found that this algorithm may produce false fitting results, and thus deteriorates the GPS filtering. In some part of road sections, the highway topology information of OSM is outdated or inaccurate, because the open source OSM database is open for everyone to edit. The above examples include most frequently happened errors in our algorithms.

Therefore, detecting errors that affect localization results can be regarded as one of the potential solutions and a first step of error correction to further improve the localization performance in vehicle localization procedure. In Zinoune et al. (2014), the potential errors are monitored according to the features captured from multiple journeys on the same road. This error correction method requires a "second" journey to compare the differences in vision and localization at the same road section. In Cui et al. (2014), a "cross validation" between GPS position and marking detection is introduced. When the GPS receiver suffers from poor satellite signals, detected lane markings are used to estimate vehicle location. When the lane markings are unable to be retrieved by image processing and model fitting, GPS based vehicle trajectory is used to complete and validate the marking detection. In other publications, the error modeling and detection are embedded in the localization algorithms. For instance, a Monte Carlo Localization is adopted in Qin et al. (2012), to eject the particles off-road, the particles out of the road area are penalized with low weights. In Najjar and Bonnifait (2005), the road geometrical tendency is regarded as one of the criteria in the algorithm to remove false positioning. The vehicle positions that are not along the road geometry will be filtered.

In our research, an on-line validation method is required to correct the filtered vehicle positions on a random road section. The validation method in Zinoune et al. (2014) adjusts the vehicle coordinates effectively, but it requires a second journey, which is not suitable in our method. In Cui et al. (2014), the GPS positioning are refined according to the lane marking shape. This step has already been implemented in our lane marking based particle filter. Considering the error modeling is independent of vehicle localization algorithm, the error detection methods Qin et al. (2012); Najjar and Bonnifait (2005) that are built in the localization process don't meet our requirement. Therefore, an error modeling and detection method is demanded, which can be useful on a certain part of road, and is a supplementary to the existing vehicle localization

approach.

The proposed probabilistic error modeling in our algorithm aims to position unsuccessful vehicle localizations, which are caused by a bad lane detection, a false multi-kernel estimation localization, and a mismatching between vision inputs and map sources. The cause of these errors can be described using several criteria, such as marking detection cues, third marking cues, and localization fitting cues. When the errors are defined and the error sources are represented, it is possible to evaluate and model these errors using probabilistic methods. A confidence of each cue is described related to corresponding error models, which can represent the probability level of the error or the error cue. To determine if an error occurs, it is necessary to estimate the thresholds of the error confidences according to a set of benchmark data. The thresholds to determine errors are decided by ROC curve Fawcett (2006). In the following statement, the errors are introduced and explained throughout example scenarios at first. Then the reasons why these errors happen are discovered. Thus the cues to reflect these errors are enumerated. Finally, the errors that cause questionable vehicle localization are evaluated, the indicator is represented and hence the error model is created. The work in this chapter has been published in Lu et al. (2014c).

4.2 Model Description

A localization result which is far away from the vehicle position benchmark is caused by certain errors. For instance, if the GPS is out of work, unusual vehicle positions will be produced, and the vehicle localization algorithm will produce vehicle coordinates with large errors. Several common mistakes can strongly impact marking based vehicle localization. Three of these usual errors are proposed, which are **false lane marking detection results**, **bad marking fitting in vehicle localization**, and **inaccurate open source map database**. We noticed that these three factors are often happened errors caused in vehicle localization. For example, in an experiment on one scenario of KITTI database, 82 of totally 1170 localization results have been found problematic results. Among these 82 detections, 34 (41.5%) of them are caused by marking fitting in vehicle localization, about 28 (34.2%) errors are inaccurate open source map database, 11 (13.4%) is attributed to false lane marking detection, while the rest 9 (11.0%) are caused by other errors. We think that these errors can be detected with respect to multi-cue based indicators. For instance, an improper lane detection can be monitored by the lane detection confidence $conf_{ld}$, which indicates the quality of the lane marking detection defined in Eq. 2.37.

False lane marking detection result means an inaccurate vision based lane marking detection. This fault is caused according to unclear vision frames with few marking feature pixels, or too many noise pixels produced in image procession stage. A bad

Table 4.1: *Causes, cues, indicators, and modeling of errors.*

Error:	Lane detection error
Error model:	Lane detection model
Cause:	Improper vision-based marking detection
Cue:	Lane marking confidence
Indicator:	$conf_{ld}$ in Eq. 4.6
Modeling:	Algorithm 4.2.6
Error:	Vehicle localization error
Error model:	Vehicle localization model
Cause:	Questionable MKE based fitting in localization
Cue:	MKE descriptor
Indicator:	$conf_{ft}$ in Eq. 4.17
Modeling:	Compare $conf_{ft}$ and $\overline{conf_{ft}}$
Error:	Inaccurate map database
Error model:	Map topology model
Cause:	Incongruent lane number from OSM and camera vision
Cue:	Third lane markings
Indicator:	$conf_{trd}$ in Eq. 4.18
Modeling:	Algorithm 4.2.7

vision based marking result apparently doesn't fit well with corresponding map based markings from OSM. This error is explained in Section 4.2.1.

In Chapter 3, MKE method is used to fit vision based markings and map based markings. In some situations of localization step, the improved MKE method may not fit the current lane markings from camera to the corresponding lane markings from OSM. The analysis and modeling of this error are stated in Section 4.2.2.

Two parts are included in inaccurate map database, inaccurate road location (nominated as "position accuracy" in Ather (2009)), which means that the coordinates of the nodes from the way are inaccurate, and wrong map topology (named as "thematic accuracy" in Ather (2009)), which means that the properties in the current cell are inaccurate, such as a wrong lane number $v_i^{way,lanes}$. Inaccurate road location happens frequently, but is difficult to evaluate at present. Although this error is not easy to discover, the analysis is represented for further research in Section 4.2.4. Map topology error can be detected using the third lane marking cue. So the error model of map topology can be created using third marking cue is presented in Section 4.2.3.

The indicators of lane detection error model, localization error model and map error model are estimated. Error occurrences are decided according to methods based on the following models. The analysis on probabilistic error modeling is summarized as Table 4.1.

4.2.1 Error Model of Lane Detection

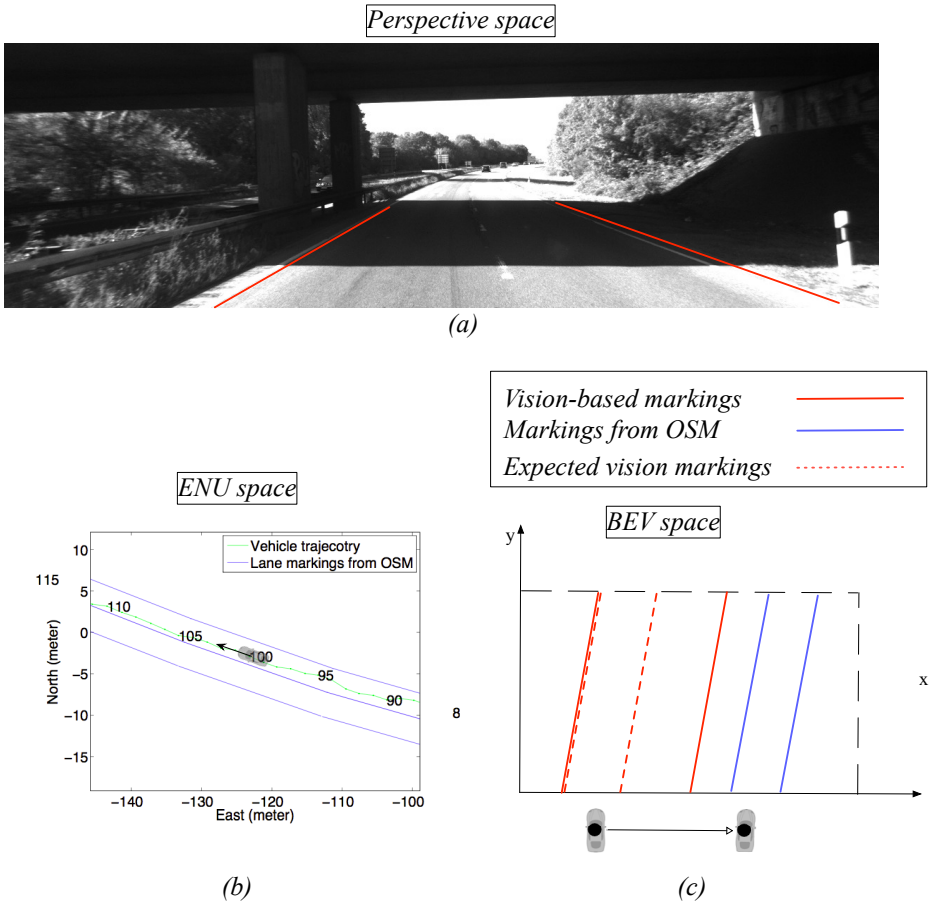


Figure 4.1: *Example of lane detection error. (a) Land detection in perspective space; (b) vehicle position in ENU space; (c) Lane markings from vision detection and OSM in BEV space.*

Improper vision-based marking detection leads to an erroneous MKE fitting of map-based lane marking, and thus, an inaccurate localization. Fig. 4.1 illustrates an example of lane detection error. In Fig. 4.1a, vision based lane markings are detected as red lines. Because of the shadow caused by the tunnel, the left marking of current cell and the right marking of the neighborhood cell are detected by mistake, instead of the pair of lane markings at current lane. Fig. 4.1b shows the map topology and vehicle fix in ENU space, the vehicle is driving in a two lane road section (green trajectories), heading to southeast, lane markings are represented as blue lines. In Fig. 4.1c, detected lane markings (red solid lines) and the map based markings (blue lines) are projected into BEV coordinate. Compared to the expected vision based markings (red dashed lines), the detected markings (red solid lines) are much wider than the map based markings. It is obvious that the pixels from two different sources are apparently unable to match. If the left side map-based marking is fitted to the left side vision-based marking, a reasonable localization is derived by coincidence. If the right side map-based marking is fitted to the right side vision-based marking, a large error in vehicle localization is

produced.

Obviously, this error is caused by a bad lane marking detection, so cues related to this error are the estimated lane marking models (c^*, d^*, e^*) and the binary images I_{blob} from image processing in Chapter 2. With the detected marking models and the binary images, lane detection error can be expressed via lane detection confidence $conf_{ld}$. When a marking (c^*, d^*, e^*) is detected from image I_{blob}, $conf_{ld}$ is employed to quantify an estimated marking, which can be a meaningful criterion to discover lane detection error. To this end, a set Φ_{ld}^{conf} of pixels in output image I_{blob} of image processing stage with higher contribution to the detected markings (c^*, d^*, e^*) are defined as follows:

$$\Phi_{ld}^{conf} = \left\{ \left(x_i^{bev}, y_i^{bev} \right) | G_{pi} \left(c^*, d^*, e^*, x_i^{bev}, y_i^{bev} \right) \geq Th_{G_{pi}} \right\}, \tag{4.1}$$

where $i = 1, ..., n_{ld}^{conf}$, $Th_{G_{pi}}$ is confidence threshold for contributed pixels, and n_{ld}^{conf} is the total number of high contributed pixels in image I_{blob}.

Then the image I_{blob} is divided into 4 average blocks along y-axis as Fig. 4.2. The set Φ_l^{ld} of pixels in block l is defined as:

$$\Phi_l^{ld} = \left\{ \left(x_i^{bev}, y_i^{bev} \right) | \left(x_i^{bev}, y_i^{bev} \right) \in \Phi_{ld}^{conf}, \frac{M_{bev}}{4} l \leq y_i^{bev} < \frac{M_{bev}}{4} (l+1) \right\}, \tag{4.2}$$

where $l_{conf} = 1, 2, 3, 4,\ i = 1, ..., n_l^{ld}$, n_l^{ld} is the number of high contributed pixels in block l.

As in Fig. 4.2, the set of total contributed pixels:

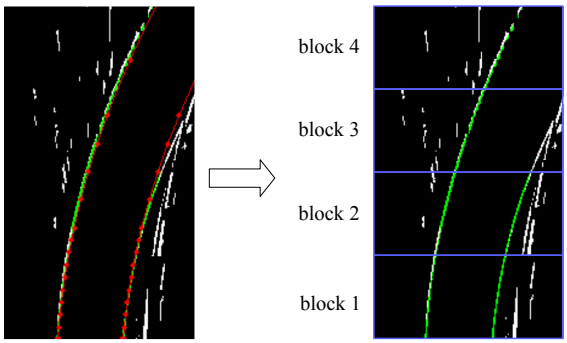
$$\Phi_{ld}^{conf} = \bigcup_{l=1}^{4} \Phi_l^{ld}, \tag{4.3}$$

is the union of the contributed pixel sets of all the 4 blocks, and the total number of contributed pixels:

$$n_{ld}^{conf} = \sum_{l=1}^{4} n_l^{ld}, \tag{4.4}$$

is the sum of the contributed pixel numbers in all the 4 blocks.

In an ideal case, the marking model crosses the whole BEV image I_{blob}, the high contributed pixels are averagely distributed along y axis. Meanwhile, the more high contributed pixels support this model, the more reliable this model is. For instance, in Fig. 4.2, the left marking model owns almost all the pixels of left marking in I_{blob} as its high contributed pixels. What's more, these green pixels scatter in all the 4 blocks. Therefore, an indicator concerning both the number of high contributed pixels and the distribution of these pixels is designed. For the total number of contributed pixels, the



In block l

Φ_l^{ld}: high contributed pixels (green pixels)

n_l^{ld}: total pixel number of Φ_l^{ld}

p_l^{ld}: confidence of lane detection in current block

In entire image

Total contributed pixels $\Phi_{ld}^{conf} = \sum_{l=1}^{4} \Phi_l^{ld}$

Total number of contributed pixels $n_{ld}^{conf} = \sum_{l=1}^{4} n_l^{ld}$

Figure 4.2: *Contributed pixels in a BEV image.*

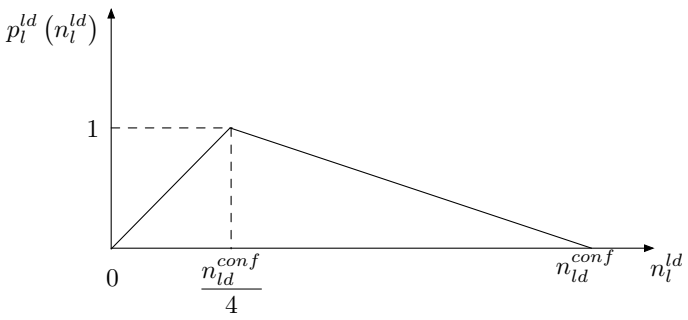


Figure 4.3: *The probability of block l .*

parameter n_{ld}^{conf} directly represents the sum of useful pixels. For the distribution of these pixels, a probability p_l^{ld} of block l is introduced as:

$$p_l^{ld}\left(n_l^{ld}\right) = \begin{cases} \frac{4 \cdot n_l^{ld}}{n_{ld}^{conf}} & \left(0 \le n_l^{ld} < \frac{n_{ld}^{conf}}{4}\right) \\ -\frac{4 \cdot n_l^{ld}}{3 \cdot n_{ld}^{conf}} + \frac{4}{3} & \left(\frac{n_{ld}^{conf}}{4} \le n_l^{ld} < n_{ld}^{conf}\right) \end{cases}. \tag{4.5}$$

The curve of $p_l^{ld}\left(n_l^{ld}\right)$ is represented as Fig. 4.3. When the pixel number in a block n_l^{ld} is one quarter of the total pixel number n_{ld}^{conf}, the probability $p_l^{ld}\left(n_l^{ld}\right)$ increases to the peak. In other words, when the contributed pixels are dispersed averagely in these four blocks, the sum of four probabilities $\sum_{l=1}^{4} p_l^{ld}\left(n_l^{ld}\right)$ reaches the maximum.

When the probabilities in all the blocks are determined, the error model is constructed. The confidence of a detection can be represented as:

$$conf_{ld} = \frac{n_{ld}^{conf}}{4} \sum_{l=1}^{4} p_l^{ld}. \tag{4.6}$$

In Eq. 4.6, $conf_{ld}$, takes the number of contributed pixels ($n_{ld}^{conf}/4$ in Eq. 4.6) and the distribution of these pixels ($\sum_{l=1}^{4} p_l^{ld}$ in Eq. 4.6) into account. A detection result which has more contributed pixels and whose contributed pixels are more equally distributed earns a higher confidence.

Algorithm 4.2.6 Error modeling of lane detection

Input: - Marking model (c_l^*, d_l^*, e_l^*) and (c_r^*, d_r^*, e_r^*)
- binary image I_{blob}
Output: error modeling of lane detection

for $(i = 0; i \leq I_{blob}.rows; i++)$ **do**
  for $(j = 0; j \leq I_{blob}.cols; j++)$ **do**
    if $G_{pi}(c^*, d^*, e^*, j, i) \geq Th_{G_{pi}}$ **then**
      ▶ $n_{ld}^{conf}++$
      ▶ record (j, i) to Φ_{ld}^{conf}
      if $i < \frac{1}{4} I_{blob}.cols$ **then** ▷ block 1
        ▶ $n_1^{ld}++$
        ▶ record (j, i) to Φ_1^{ld}
      else
        if $\frac{1}{4} I_{blob}.cols \leq i < \frac{1}{2} I_{blob}.cols$ **then** ▷ block 2
          ▶ $n_2^{ld}++$
          ▶ record (j, i) to Φ_2^{ld}
        else
          if $\frac{1}{2} I_{blob}.cols \leq i < \frac{3}{4} I_{blob}.cols$ **then** ▷ block 3
            ▶ $n_3^{ld}++$
            ▶ record (j, i) to Φ_3^{ld}
          else
            if $\frac{3}{4} I_{blob}.cols \leq i < I_{blob}.cols$ **then** ▷ block 4
              ▶ $n_4^{ld}++$
              ▶ record (j, i) to Φ_4^{ld}
            end if
          end if
        end if
      end if
    end if
  end for
end for
for $l = 1; l \leq 4; l++$ **do**
  ▶ calculate p_l^{ld} according to Eq. 4.5
end for
▶ calculate $conf_{ld}$ according to Eq. 4.6
▶ calculate d_{vision} according to Eq. 4.7
if $conf_{ld} \geq \overline{conf_{ld}}$ **then** ▷ condition 1
  if $\left| d_{vision} - d_{i,j}^{seg,lane} \right| < \Delta_{width}^{ld}$ **then** ▷ condition 2
    ▶ **A lane detection error is determined.**
  end if
end if

The other error indicator is the lane width d_{vision} derived from detected left and right marking model (c_l^*, d_l^*, e_l^*) and (c_r^*, d_r^*, e_r^*) from Eq. 2.22 and Eq. 2.32. Width d_{vision} can be computed using zero order components c_l^* and c_r^* as:

$$d_{vision} = \frac{|c_r^* - c_l^*|}{T_{bev2enu}}, \tag{4.7}$$

where $T_{bev2enu}$ is metric coefficient from BEV space to ENU space.

A lane detection error is modeled based on $conf_{ld}$ from Eq. 4.6 and lane width $d_{i,j}^{seg,lane}$ defined in Eq. 3.17. An unsuccessful detection contains two conditions. One condition is that $conf_{ld}$ is lower than a threshold $\overline{conf_{ld}}$. The second condition relies on the differences Δ_{width}^{ld} between the vision-based lane width and the lane width provided by OSM. When both conditions are satisfied, a lane detection error is found. The algorithm of this error modeling is demonstrated as Algorithm 4.2.6.

4.2.2 Error Model of Localization

Multi-kernel estimation based marking fitting plays an important role in the whole vehicle localization method. A successful fitting example is shown in Fig. 4.4. In Fig. 4.4a, the current lane markings are detected in perspective space. The marking models are then transmitted to BEV space as red lines in Fig. 4.4b, as well as the markings from open source map database as blue lines. The multi-kernel based estimation can help to derive the motion vector from vision-based lane markings to map-based lane markings. As shown in Fig. 4.4c, the markings from two different sources fit well, which verifies a successful localization.

However, abnormal situations may happen in some conditions. For instance, the vision-based left marking matches the map-based right marking, rather than map-based left marking, as depicted in Fig. 4.5. The camera view is shown in Fig. 4.5a, the lane markings of current lane are detected as red dots. Fig. 4.5b is the projection of both lane marking sources on BEV space before localization. The error occurred in Fig. 4.5c, the detected left marking (red line) fits the right side marking from map (blue line), while the expected localization is to fit the detected left (right) marking to map based left (right) markings. In this paradigm, the estimated motion vector $(\triangle x_{loc}^*, \triangle \theta_{loc}^*)$ leads to a error vehicle localization.

The reason of this error is in road environment, the distance between the left and right markings from vision differs from the lane width from map database. For instance, assuming that lane width difference between vision-based markings and map-based markings is 6 pixels in BEV space, which is 6 dm in reality. This condition is shown as Fig. 4.6. If the interval is 6 dm, the ideal MKE fitting result is an interval of 3 dm on both left and right side, depicted as Fig. 4.6b. However, in reality, the MKE fitting

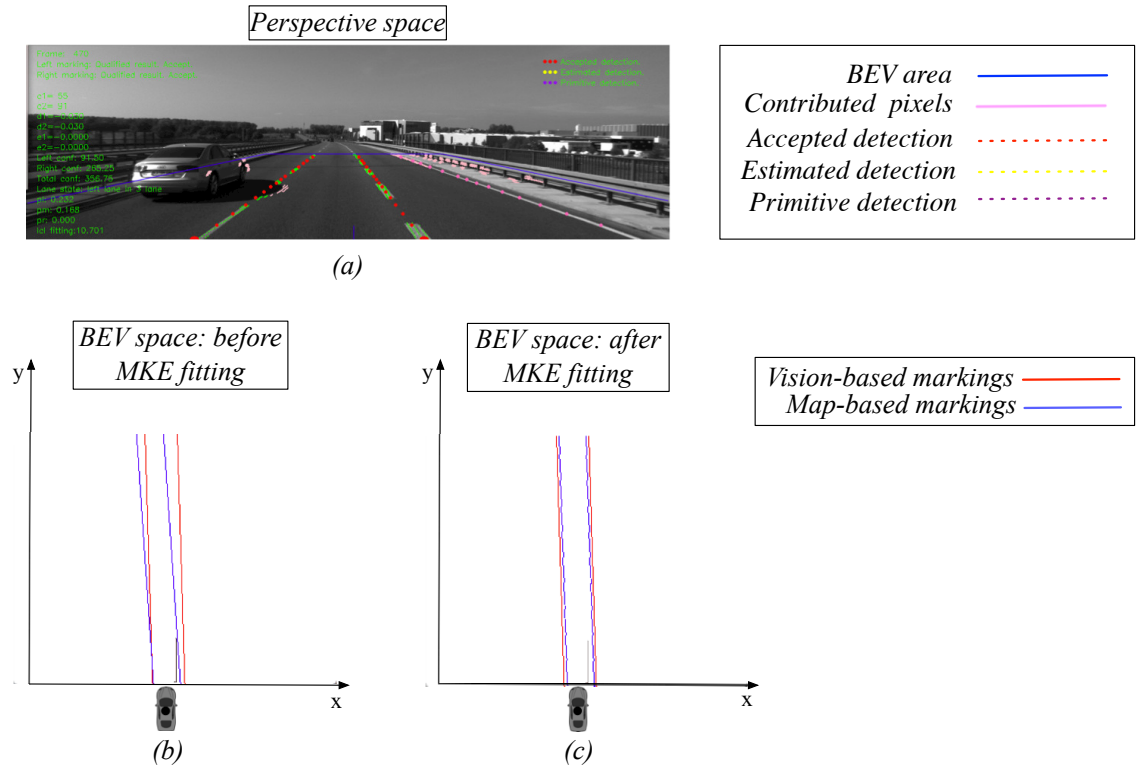


Figure 4.4: *Example of successful lane marking fitting. (a) Lane detection in perspective space; (b) the markings before MKE fitting; (c) the markings after MKE fitting.*

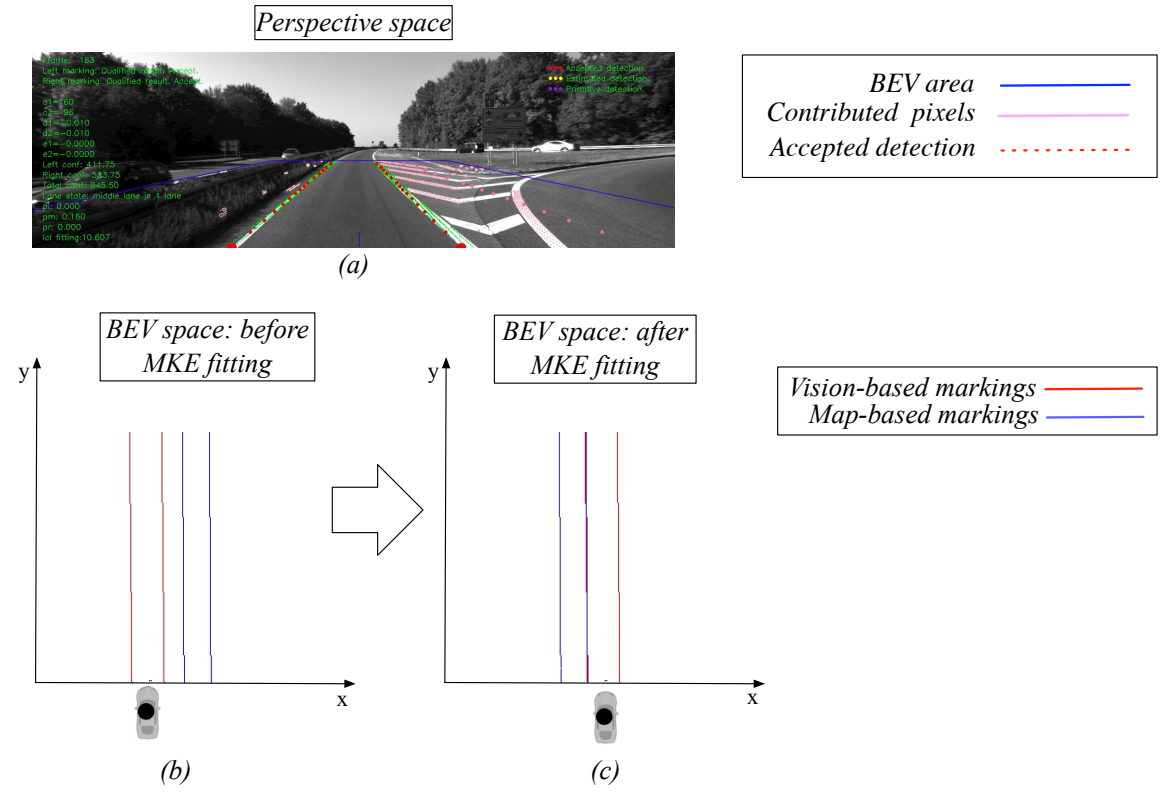


Figure 4.5: *Example of localization error. (a) Lane detection in perspective space; (b) the markings before MKE fitting; (c) the markings after MKE fitting.*

result is more likely as Fig. 4.6c, on one side the difference is 0 dm, while on the other side it is 6 dm.

A numerical experiment was implemented to verify that the fitting result in Fig. 4.6c has higher possibility than that in Fig. 4.6b. Assume that $(x_{mm}^{em}, y_{mm}^{em})$ is a single pixel on map-based markings, vision-based marking model (c, d, e) is fixed, so the distances between $(x_{mm}^{em}, y_{mm}^{em})$ and the corresponding marking model are $\triangle x^{em}$ and $\triangle y^{em}$on x and y axis respectively. The MKE descriptor from Eq. 2.23 of $\triangle x^{em}$ and $\triangle y^{em}$ can be expressed as:

$$G_{pi}\left(\triangle x^{em}, \triangle y^{em}\right) = \int_{-\infty}^{+\infty} K_x^{ori}\left(\triangle x^{em}\right) K_y^{ori}\left(\triangle y^{em}\right) dy, \tag{4.8}$$

where

$$K_y^{ori}\left(\triangle y^{em}\right) = \frac{1}{\sqrt{2\pi\left(\sigma_y^{ori}\right)^2}} exp(-\frac{\left(\triangle y^{em}\right)^2}{2\left(\sigma_y^{ori}\right)^2}), \tag{4.9}$$

$$K_x^{ori}\left(\triangle x^{em}\right) = \frac{1}{\sqrt{2\pi\left(\sigma_x^{ori}\right)^2}} exp(-\frac{\left(\triangle x^{em}\right)^2}{2\left(\sigma_x^{ori}\right)^2}). \tag{4.10}$$

To simplify the computation complexity, Eq. 4.8 is discretized as:

$$G_{pi}(\triangle x^{em}, \triangle y^{em}) = \sum_{\triangle y = y_{mm}^{em} - 2\sigma_{gpi}}^{y_{mm}^{em} + 2\sigma_{gpi}} .K_x^{ori}(\triangle x^{em}) K_y^{ori}(\triangle y^{em} + \triangle y). \tag{4.11}$$

Set $\sigma_{gpi} = 0$, $G_{pi}(\triangle x^{em}, \triangle y^{em})$ can be further simplified as:

$$G_{pi}\left(\triangle x^{em}\right) = \frac{K_x^{ori}\left(\triangle x^{em}\right)}{\sqrt{2\pi\left(\sigma_y^{ori}\right)^2}}. \tag{4.12}$$

If set $\sigma_x^{ori} = 2$ and $\sigma_y^{ori} = 2$, $K_x^{ori}\left(\triangle x^{em}\right)$ is depicted as Fig. 4.6a. When substitute $\triangle x^{em} = 0dm, \pm 3dm, -6dm$ into Eq. 4.12, the value of $G_{pi}\left(0\right)$, $G_{pi}\left(\pm 3\right)$ and $G_{pi}\left(-6\right)$ can be calculated. The sum MKE descriptor of left and right markings in Fig. 4.6b is:

$$G_{pi}^{emb} = G_{pi}\left(3\right) + G_{pi}\left(-3\right) = 0.0250, \tag{4.13}$$

while the total descriptor in Fig. 4.6c is:

$$G_{pi}^{emc} = G_{pi}\left(0\right) + G_{pi}\left(-6\right) = 0.0398. \tag{4.14}$$

The descriptor of erroneous condition is greater than ideal condition. Therefore, when

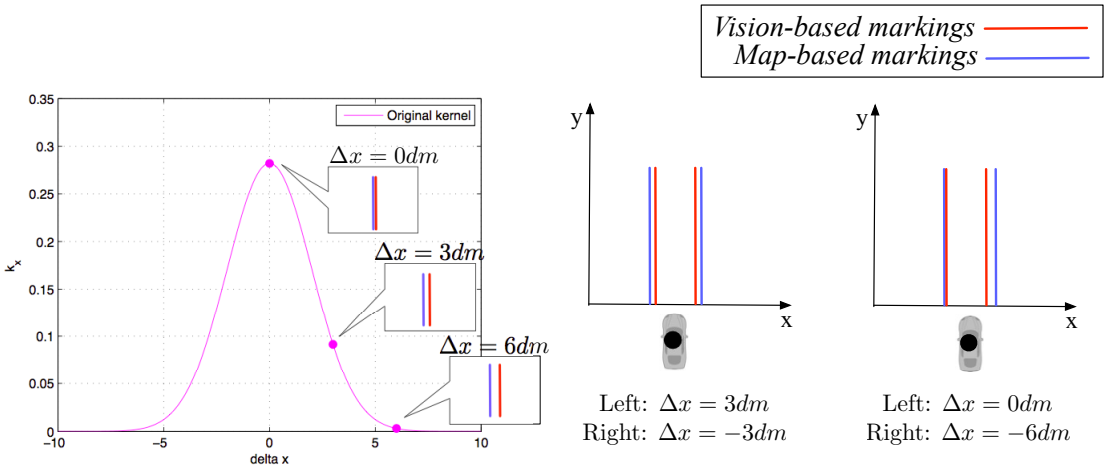


Figure 4.6: *Width difference situation. (a) Reason of localization error; (b) an ideal MKE fitting in width difference situation; (c) an erroneous MKE fitting*

a lane width difference occurs, the fitting of vehicle localization tends to derive an erroneous situation.

The most obvious characteristic of erroneous situations in MKE localization is the different matching level of left and right side. Therefore, to derive the fitting probability of left and right marking separately is an effective way to diagnose localization error. At first, when $\triangle x^*_{loc}$ and $\triangle \theta^*_{loc}$ are computed, the set of transmitted map-based lane marking pixels on left and right side $S_{l,tra}$ and $S_{r,tra}$ from Eq. 3.68 and Eq. 3.69 are determined. The fitting confidence of left and right side $p^{em,loc}_{left}$ and $p^{em,loc}_{right}$ can be represented as:

$$p^{em,loc}_{left} = \frac{1}{n_{l,tra}} \sum_{\left(x^{bev}_i, y^{bev}_i\right) \in S_{l,tra}} w_{mke}\left(x^{bev}_i, y^{bev}_i\right) \cdot G_{pi}\left(x^{bev}_i, y^{bev}_i\right), \tag{4.15}$$

$$p^{em,loc}_{right} = \frac{1}{n_{r,tra}} \sum_{\left(x^{bev}_i, y^{bev}_i\right) \in S_{r,tra}} w_{mke}\left(x^{bev}_i, y^{bev}_i\right) \cdot G_{pi}\left(x^{bev}_i, y^{bev}_i\right), \tag{4.16}$$

where $n_{l,tra}$ and $n_{r,tra}$ are the total number of pixels in $S_{l,tra}$ and $S_{r,tra}$ respectively. $p^{em,loc}_{left}$ and $p^{em,loc}_{right}$ are regarded as the level how the two marking sources are fitted. Marking fitting confidence $conf_{ft}$ is used to detect this kind of failure. When $p^{em,loc}_{left}$ and $p^{em,loc}_{right}$ are derived, error confidence is defined according to the following equation:

$$conf_{ft} = \frac{\left| p^{em,loc}_{left} - p^{em,loc}_{right} \right|}{p^{em,loc}_{left} + p^{em,loc}_{right}}. \tag{4.17}$$

When $conf_{ft}$ tends to 0, it implies a high quality marking fitting, because the left and right fitting confidences $p_{left}^{em,loc}$ and $p_{right}^{em,loc}$ are almost equal. If $conf_{ft}$ is higher than threshold $\overline{conf_{ft}}$, a localization failure is identified.

4.2.3 Error Model of Map Topology

A map error here stands for a fault lane number assigned to a road segment in OSM. Normally, the lane number $n_{i,j}^{seg,lane}$ from map database matches the road situation from the vision. For instance, in Fig. 4.7a, the vehicle is in a 2-lane condition from vision. Meanwhile, according to the vehicle position, the current lane index is determined. The lane number of current road cell is derived from OSM, which is $n_{i,j}^{seg,lane} = 2$. The lane numbers from two sources are the same. However, in Fig. 4.8a, the lane number is two, one current lane and one branch. But the lane number offered from OSM is 1. In this condition, an error of map topology occurs.

Third lane markings, can be used to determine the multi-lane situations, and in advance, model the map topology error. The programming algorithm is demonstrated as Algorithm 4.2.7. At, first, the image after second order filter I_{xx} is selected as the input image because the image after cell based blob algorithm I_{blob} tends to erase some features of third markings. The third lane marking model $(c_{trd}, d_{trd}, e_{trd})$ is estimated according to the current lane marking model (c^*, d^*, e^*). For parameter d_{trd} and e_{trd}, loosen offsets $\triangle d_{trd}$ and $\triangle e_{trd}$ are added as $d_{trd} \in [d^* - \triangle d_{trd}, d^* + \triangle d_{trd}]$ and $e_{trd} \in [e^* - \triangle e_{trd}, e^* + \triangle e_{trd}]$. For parameter c_{trd}, the lane width d_{lane} is considered into the estimated range, shown as blue lines in Fig. 4.7b. The starting point of the left/right third marking is restricted according to the blue range produced according to the detected left/right marking. So the zero order parameter of left third marking is determined as $c_{trd,l} \in [(c_l^* - d_{lane}) - \triangle c_{trd}, (c_l^* - d_{lane}) + \triangle c_{trd}]$, while one the right marking side $c_{trd,r} \in [(c_r^* + d_{lane}) - \triangle c_{trd}, (c_r^* + d_{lane}) + \triangle c_{trd}]$. Then the descriptor G_{pi} in Eq. 2.23 and the marking probability p_{Gpi} are introduced here to measure the lane markings. The procedure is similar as the lane detection algorithm in Chapter 2. When the third lane marking model $c_{trd}^*, d_{trd}^*, e_{trd}^*$ is derived, a confidence $conf_{trd}$ of this marking is computed as:

$$conf_{trd} = p_{Gpi}\left(c_{trd}^*, d_{trd}^*, e_{trd}^*\right). \tag{4.18}$$

The third lane markings with confidence $conf_{trd}$ greater than $\overline{conf_{trd}}$ are considered as potential third markings, shown as green markers in Fig. 4.12b and Fig. 4.12c. Barriers or other noises sometimes can cause an instantaneous fake third marking, so the markings keeping visible for a period is considered as a detected third marking. Once the third lane markings are derived, map error can be modeled logically through fusing third marking and number of lanes from OSM. For instance, when the vision

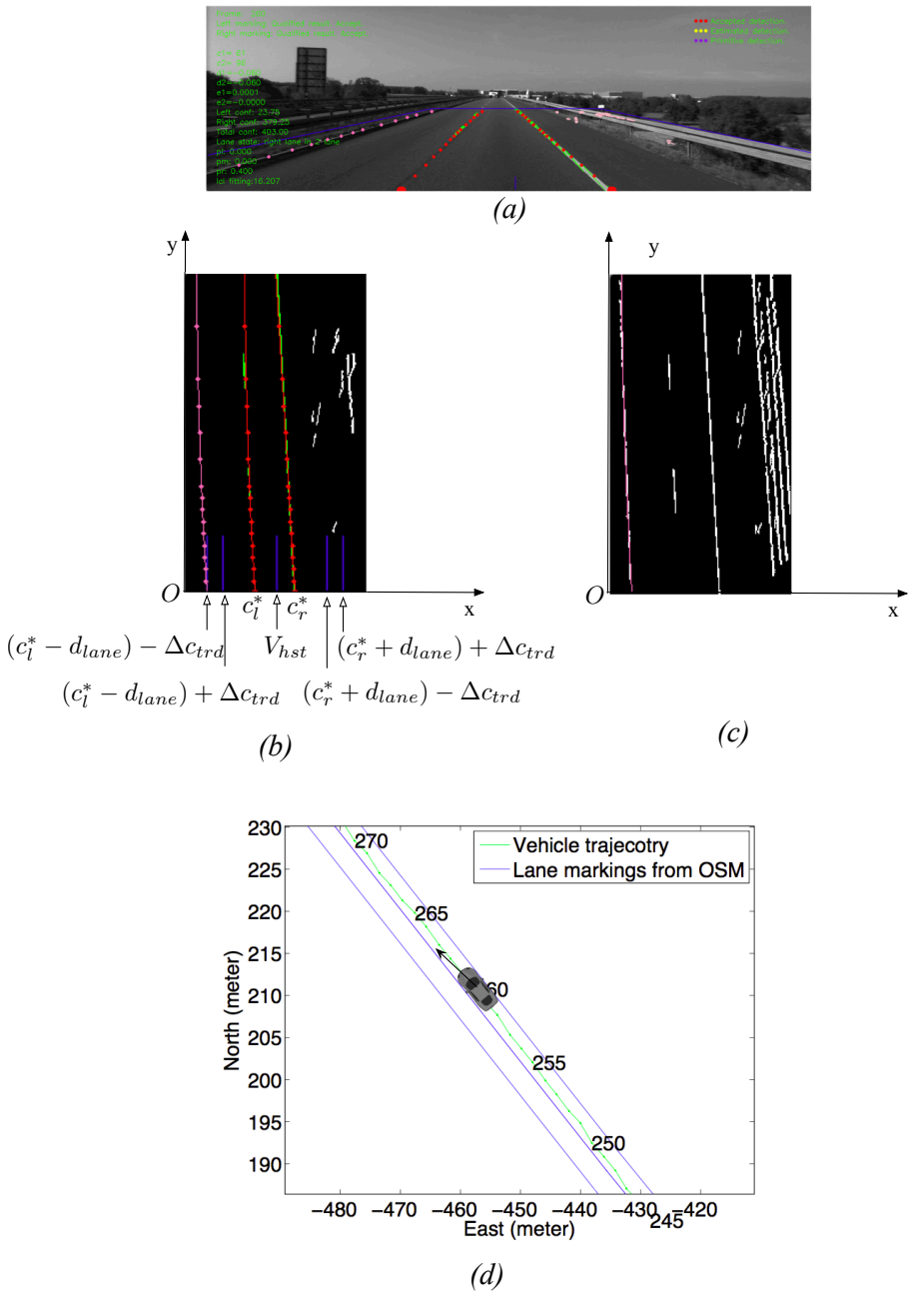


Figure 4.7: *Example of normal detection in map topology error modeling. (a) Lane detection in perspective space; (b) detected the markings of the host lane and the third marking in BEV space; (c) the third lane marking and lane marking "signatures"; (d) vehicle position in ENU space. Red curves are the markings of the host lane, while pink lines are the third lane markings.*

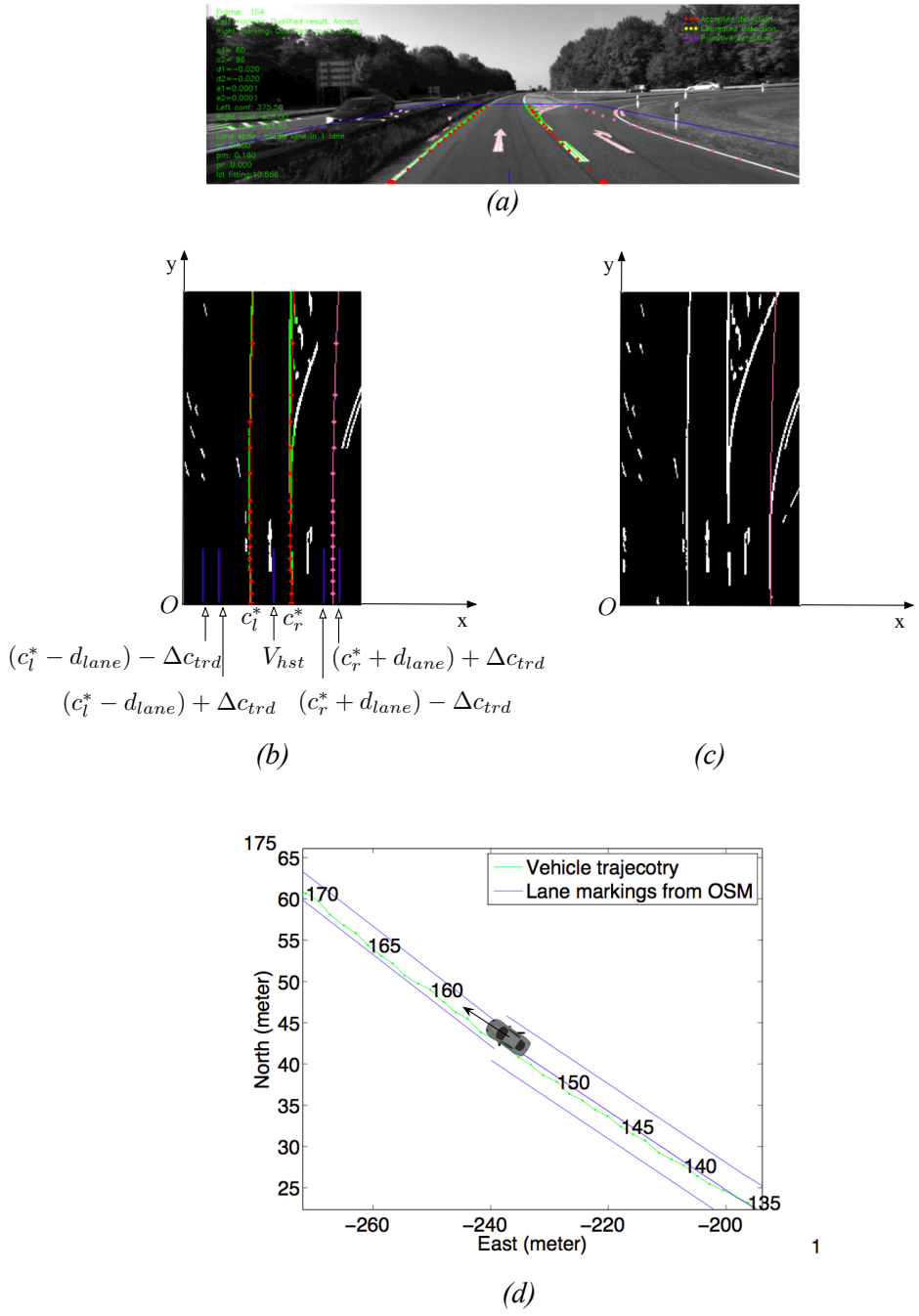


Figure 4.8: *Example of error detection in map topology error modeling. (a) Lane detection in perspective space; (b) detected the markings of the host lane and the third marking in BEV space; (c) the third lane marking and lane marking "signatures"; (d) vehicle position in ENU space. Red curves are the markings of the host lane, while pink lines are the third lane markings.*

detects a third lane marking, but the map database indicates the current section is an one-lane road section, so an error is identified. Table 4.2 explains the effect of third lane marking and lane number $n_{i,j}^{seg,lane}$ on a map topology error. Because the third lane marking is not always detected as the current lane markings, it is not asserted that a third lane marking is detected for sure. When no third lane marking is detected, the rover is possible in a 1-lane, 2-lane or 3-lane condition, so no error is alarmed. When either side of third marking is confirmed, the lane number of current location is at least 2, so if $n_{i,j}^{seg,lane} = 1$, an error of map topology will be reported. When third markings on both sides are derived, the current lane number equals to or is more than 3, so if the lane number from OSM $n_{i,j}^{seg,lane} \leq 2$, it is a map topology error.

Algorithm 4.2.7 Error model of map topology

Input: - binary BEV image I_{xx}

Output: error modeling of map topology

for $e = e^* - \triangle e_{trd}; e <= e^* + \triangle e_{trd}; e = e + e_{reso}^{mke}$ **do**
  for $d = d^* - \triangle d_{trd}; d <= d^* + \triangle d_{trd}; d = d + d_{reso}^{mke}$ **do**
    for $c = (c^* \pm d_{lane}) - \triangle c; c <= (c^* \pm d_{lane}) + \triangle c; c = c + 1$ **do**
    ▷ when calculating left marking use "−", when calculating right marking use "+"
      for $i = 0; i < I_{xx}.row; i++$ **do**
        for $j = c + d \cdot i + e \cdot i^2 - 3 \cdot \sigma_{gpi}^{ipv}; j < c + d \cdot i + e \cdot i^2 + 3 \cdot \sigma_{gpi}^{ipv}; j++$ **do**
                                                                ▷ meet Eq. 2.31
          if $I_{xx}(i,j) \neq 0$ **then**
            ▶ calculate $G_{pi}(c,d,e,i,j)$ ▷ using Eq. 2.23 or Eq. 2.30
            ▶ $p_{Gpi}(c,d,e) += w_{mke}(i,j) \cdot G_{pi}(c,d,e,i,j)$
          end if
        end for
      end for
    end for
  end for
end for
▶ $c_{trd}^*, d_{trd}^*, e_{trd}^* = \arg\max_{c,d,e} p_{Gpi}(c,d,e)$
▶ $conf_{trd} = p_{Gpi}(c_{trd}^*, d_{trd}^*, e_{trd}^*)$ ▷ get the third lane marking confidence
if $conf_{trd} \geq \overline{conf_{trd}}$ **then**
  ▶ Find a third marking
  if this third marking lasts for $N_{tra}^{em,map}$ frames **then**
    ▶ determine this third marking
    ▶ judge a map topology error using Table 4.2
  end if
end if

Table 4.2: *Logical relationship of third lane marking and number of lanes.*

Third marking state		lane number $n_{i,j}^{seg,lane}$		
Left marking	Right marking	1	2	3
Not detected	Not detected	Normal	Normal	Normal
Not detected	Detected	Map topology error	Normal	Normal
Detected	Not detected	Map topology error	Normal	Normal
Detected	Detected	Map topology error	Map topology error	Normal

4.2.4 Discussion

Inaccurate map database includes two potential errors, one is the error of map topology, which is introduced above, and the other is the error of inaccurate road location. Fig. 4.9 depicts a comparative example of this inaccurate road location. In the ENU space of Fig. 4.9a, the lane markings (blue lines) are constructed from OSM, forming a 2-lane road scene, the GPS positions (green dots) are in the right lane; while in corresponding perspective space (camera view), the vehicle is in a 2-lane road section (a third lane on the right side is a lane for emergency, so this lane is not counted as a normal lane). A scene with error is shown as Fig. 4.9b. From the perspective view, the vehicle is in the right lane of a 3-lane section. However, in ENU space, the GPS positions of this part is not in the right lane, even out of the road, which is obviously a mismatching between lane marking and GPS.

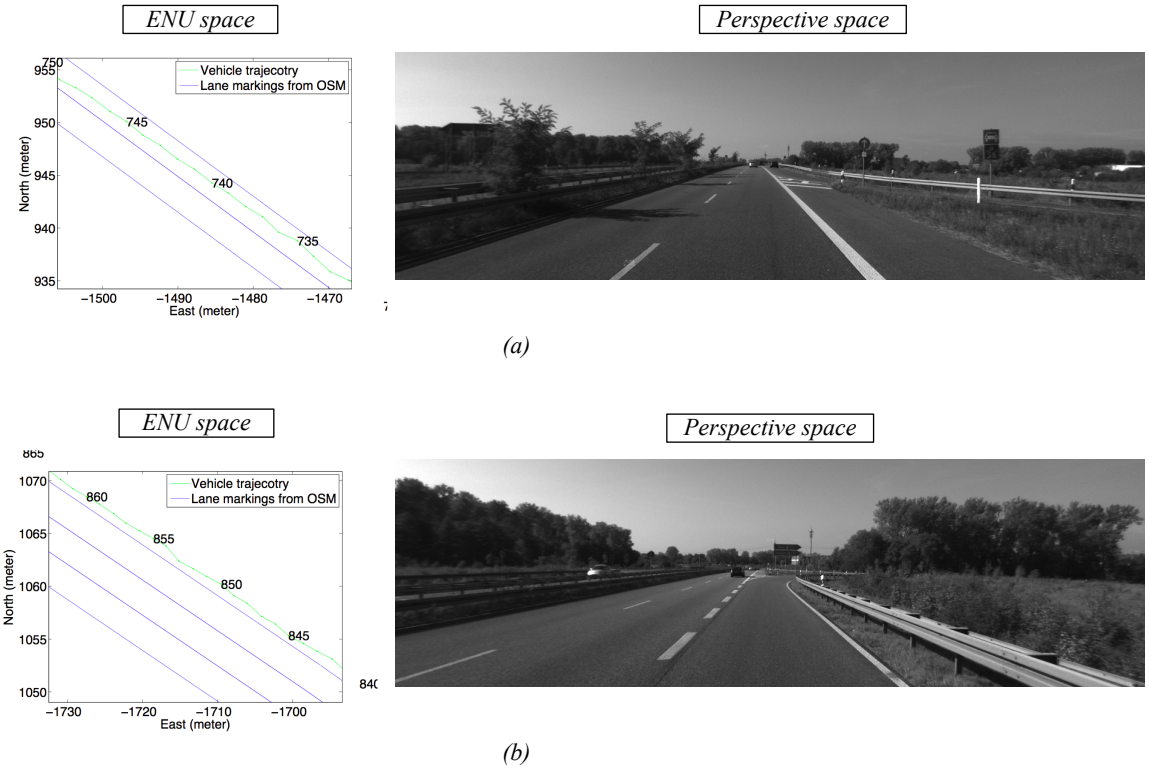


Figure 4.9: *Example of inaccurate road location. (a) A normal scenario between lane markings and GPS; (b) a mismatching scenario between lane markings and GPS.*

Table 4.3: *Confusion matrix of $conf_{ld}$.*

		with threshold $\overline{conf_{ld}}$	
		$conf_{ld} \leq \overline{conf_{ld}}$	$conf_{ld} > \overline{conf_{ld}}$
with ground truth	an error	True Positive	False Negative
	not an error	False Positive	True Negative

This mismatching happens because the lane markings and vehicle coordinates come from different data source. Lane marking map is reconfigured based on OSM database. GPS data is from a high-precision navigation system, provided by KITTI database Geiger et al. (2013). Therefore, at least one data source contains error measurements.

At OSM side, it is announced in a quality analysis in Ather (2009), OSM and OS MasterMap own 80% overlap between most the road objects, which means that in these two map databases, the differences of 80% of road objects are within 1 meter. In Haklay (2010), it is reported that the average error of OSM across several domains is 5.83 meter, compared to Ordnance Survey (OS) positions, which is the map database from the national mapping agency for Great Britain. At KITTI side, the GPS benchmark position is measured by OXTS RT3003 GPS/IMU, whose position accuracy is less than 1.5 meters Oxf (2014). The comparison on position accuracy of OSM and KITTI reveals that if a mismatching occurs, it is more probable an error caused by map than GPS benchmark.

However, the error of mismatching is almost impossible to detect without high-accuracy GPS signals. So the error of mismatching is not modeled yet.

4.3 Threshold Estimation

When the confidences $conf_{ld}$ and $conf_{ft}$ are estimated, the potential error can be decided according to the thresholds $\overline{conf_{ld}}$ and $\overline{conf_{ft}}$. To determine the optimized thresholds, confusion matrices of $conf_{ld}$ and $conf_{ft}$ are constructed as Table 4.3 and 4.4 respectively, including the required confusion matrix elements: true positive (TP), false positive (FP), true negative (TN) or false negative (FN).

A higher confidence $conf_{ld}$ means a lower possibility of lane detection error, while a higher confidence $conf_{ft}$ represents a higher possibility of localization error. Therefore, the definition of matrix elements in Table 4.3 and 4.4 are different. The optimized thresholds are derived using ROC curve. The third marking threshold $\overline{conf_{trd}}$ is assigned similar as marking detection threshold $\overline{conf_{ld}}$, as demonstrated in Table 4.5. In error modeling of map topology, $conf_{trd}$ doesn't decide an error directly, so the ground

Table 4.4: *Confusion matrix of $conf_{ft}$.*

		with threshold $\overline{conf_{ft}}$	
		$conf_{ft} \leq \overline{conf_{ft}}$	$conf_{ft} > \overline{conf_{ft}}$
with ground truth	an error	False Negative	True Positive
	not an error	True Negative	False Positive

Table 4.5: *Confusion matrix of $conf_{trd}$.*

		with threshold $\overline{conf_{trd}}$	
		$conf_{trd} > \overline{conf_{trd}}$	$conf_{trd} \leq \overline{conf_{trd}}$
with ground truth	a third marking	True Positive	False Negative
	no third marking	False Positive	True Negative

truth in Table 4.5 is not the error state, but the third lane marking state.

4.4 Experimental Validation

The results of error modeling are based on the experimental results of vehicle localization in KITTI database in Chapter 3. In the previous chapter, the our algorithm is tested on one of the longest scenarios in KITTI database, the detected marking models, the confidences of the markings, the selected lane cells, the filtered vehicle positions via particle filter, the GPS corrections according to MKE and other results are used to evaluate our algorithm. In this subsection, the analysis and assessment of error modeling approach are represented on the same data scenario to further demonstrate and testify our method. The results of all three error models are analyzed respectively. The identified errors are compared with error ground truth, which is annotated manually frame by frame. The effect of each error model is evaluated throughout the number of error detections and the localization error after error modeling. The global improvement of error detection is illustrated in the end.

4.4.1 Lane Detection Error

The observed lane detection error with ground truth is illustrated in Fig. 4.10. Two cues are retrieved to discover lane detection errors: lane marking confidence, and lane width. In Fig. 4.10a, the sum of left and right lane marking confidences is illustrated as the

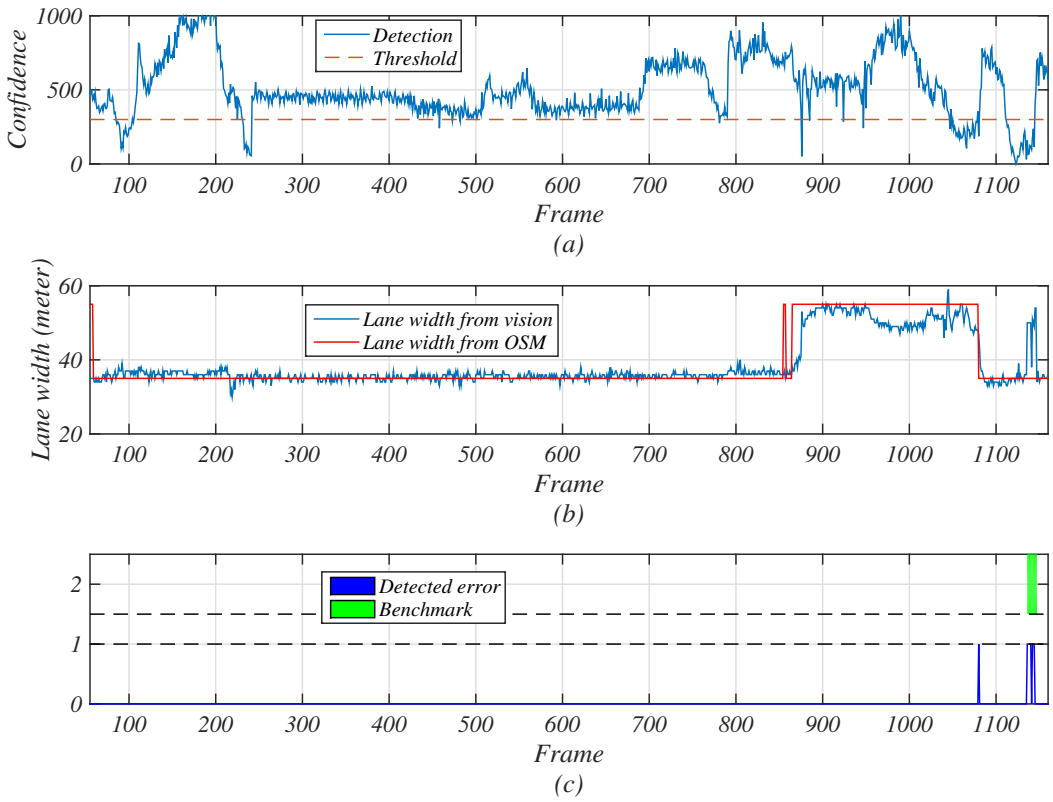


Figure 4.10: *The lane detection error. (a) Marking confidence; (b) detected lane width and map-based lane width; (c) detected error and ground truth*

blue curve. At around frame 100 to 200 and frame 800 to 1000, the average confidence is higher than that at frame 300 to 700. The reason is than at frame 100 to 200, and 800 to 1000, both the lane markings are solid lines, the lane marking "signatures" are therefore more than dashed lines at frame 300 to 700, thus the confidence is higher. The threshold of marking confidence is represented as red line in Fig. 4.10a. The confidence lower than the threshold is regarded as a potential lane detection error. In this scenario, 90.4% of the frames have successful lane detections according to the threshold. However, single marking confidence cannot judge a lane detection. At about frame 450, both lane markings are dashed lines, when not enough marking sections are captured in BEV space, the marking confidence can lower than the threshold, even the detection is correct.

So it is necessary to introduce lane width as a second cue, as shown in Fig 4.10b. The blue curve is the lane width derived from vision, while the red one is that from OSM. Both lane width curves jumped from 3.5 meter to 5.5 meter at frame 900 to 1050. At this road section, the rover drives from a straight road, whose width is 3.5 meter, to a corner ramp road, whose width is 5.5 meter. And after frame 1050, the rover drives through the ramp road to another straight highway. Notice that a peak of red curve happens at about frame 850, this is caused by the lane selection process.

To derive lane width from OSM, the host lane cell is estimated in the lane selection part of vehicle localization method. Then the width of the host cell is obtained as the lane width from map, which is red curve in Fig 4.10b. At around frame 850, the rover is at the connection of a straight cell and a corner cell. In lane selection process, the lane confidences of the straight cell and the corner cell are similar, the decision change leads to a sharp peak at frame 850. If the widths from two different sources are not agreed, and the confidence is below the qualification line, a lane detection error is confirmed. For instance, in frame 1080, a lane detection error is detected, because the lane marking confidence $conf_{ld}$ is lower than the threshold and the lane width is abnormal. Such kind of false alarm can be caused by an affected lane marker or a poor vision. At around frame 850, the lane detection algorithm works normally, regardless of the lane selection difficulty, no lane detection error is alarmed here.

4.4.2 Localization Error

The error state of vehicle localization is shown in Fig. 4.11. At first, the cue confidence of localization error is drawn. A higher confidence means a worse fitting between the markings from vision source and markings of map database. For instance, the left/right lane marking from lane detection algorithm fits to the right/left marking from OSM, the confidence of localization error cue tends to 1. This confidence is adequate to point out localization error. As an example, a threshold of 0.8 is set as the boundary of qualified localization and unqualified localization. The discovered localization errors are depicted in Fig. 4.11. The error frames match the frames with high localization error confidence. Compared to the ground truth errors, some False Positive (FP) and False Negative (FN) situations appear.

False Positive states, known as false alarms, happen in frame 550, an unusual frame is detected where no error occurred actually. The reason is that the lane widths from map source and from vision detection are different, leading to a difficulty in marking fitting.

One False Negative situation happens at the beginning of the scenario. In the first several frames, the host road cell is not discovered yet, so no lane markings from OSM is provided, only markings from vision based lane detection are available. Both the left and right multi-kernel estimation based fitting confidences are 0, because there are no marking pixels from OSM to contribute to curve fitting. According to the definition of localization error confidence in Eq. 4.17, the confidence is 0, which indicates an ideal curve fitting.

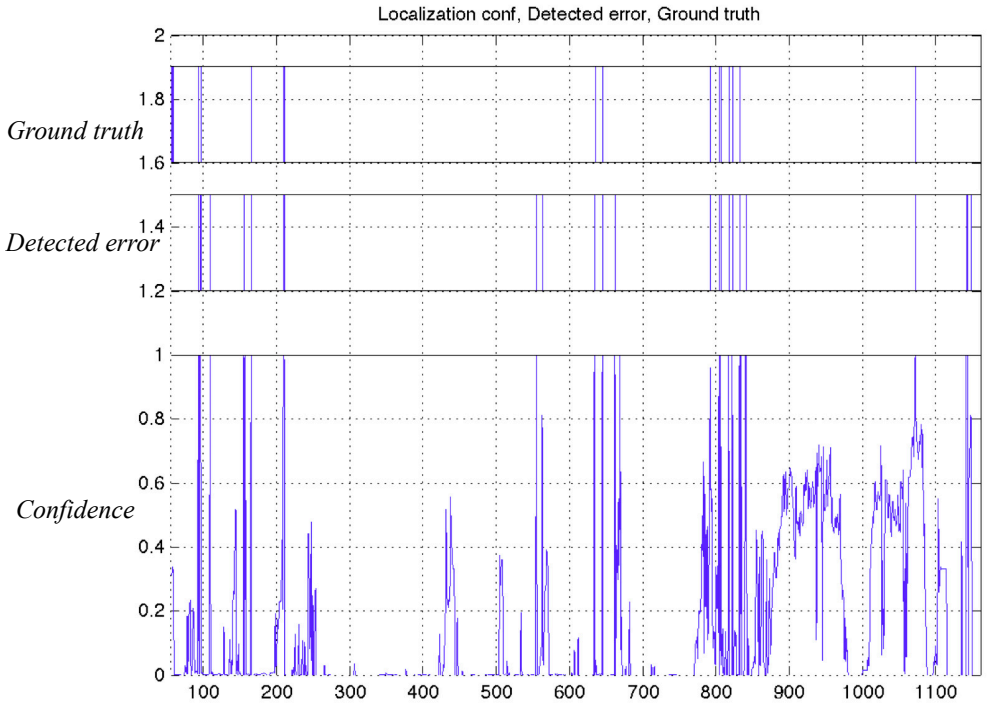


Figure 4.11: *Localization confidence and localization errors.*

4.4.3 Map Topology Error

Map topology errors are discovered according to the third lane marking, which is the marking of the neighborhood cell, as well as the cell property from OSM. The third lane markings are detected using the on-vehicle camera. The confidences of left and right third marking are shown in Fig. 4.12b and Fig. 4.12c respectively. The third lane marking detection is not as stable as host lane marking detection because: 1) a third lane marking is normally near the left or right boundary of BEV image, the marking "signatures" are not always adequate; 2) noises such as road barriers and green areas affect the detection. Therefore, to determine a third lane marking, a single marking confidence is not persuasive. The continuity of third lane marking is introduced. At first third marking threshold (assign as 400 here) is used to determine a successful third marking detection, marked as green peaks in Fig. 4.12b-c. During these successful detections, barriers and road-side trees can be estimated as third markings. Then, among the successful detections, the results lasting more than 10 frames are considered as a stable third marking. These stable markings are shown as the read peak in Fig. 4.12b and c.

The other source is the lane number provided by the OSM, which is shown as the red curve in Fig. 4.12a. In this scenario, 1-lane road, 2-lane road, and 3-lane road appear along the highway. When the third markings and lane numbers are derived, the map topology errors are judged according to the logical relationship in Table 4.2.

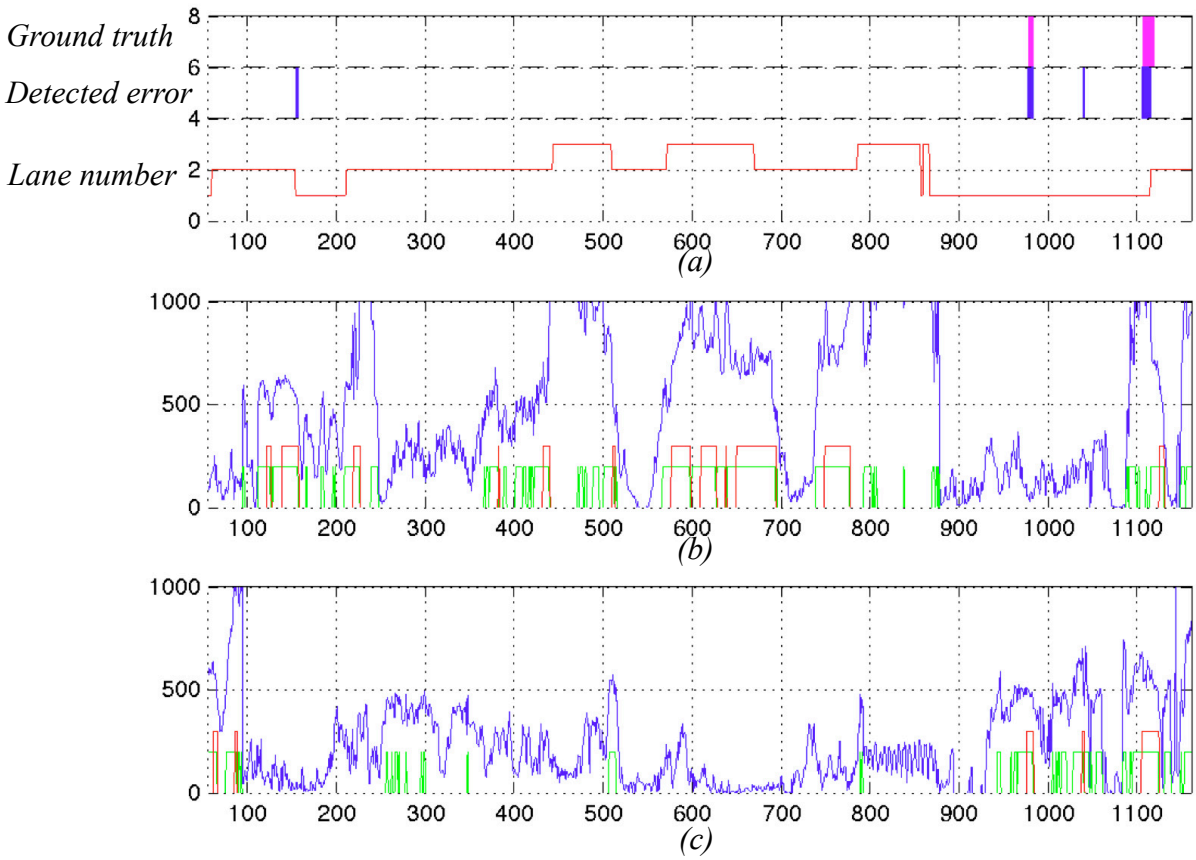


Figure 4.12: *Map error. (a) Detected map error and ground truth; (b) right third marking confidence; (c) left third marking confidence. In (b) and (c), green lines are potential third markings, red lines represent believable third marking.*

The detected map errors are depicted in Fig. 4.12a, as well as the ground truth. In Frame 980, the lane number of the current section is one, while from vision, a third lane marking on the right side indicates that the vehicle is in a two-lane road section. Third lane marking and lane number from OSM help to catch this error. From Frame 1100 to Frame 1120, the map error frames do not seem fit strictly, because the vision has discovered a third lane marking in the following two-lane section, but the vehicle itself is still in the previous one-lane section. In Frame 1040, the barrier is detected as a third lane marking by mistake, this fake third marking causes false alarm of map error.

4.4.4 Global Results of Error Modeling

Fig. 4.13 shows the localization result with the inferred errors. In Fig. 4.13a, an original localization result is represented. Blue lines are noise bounds. The cyan curve is the noised GPS position. The position processed by Particle Filter is shown as yellow curve. And the red curve is the optimized vehicle position. The Mean

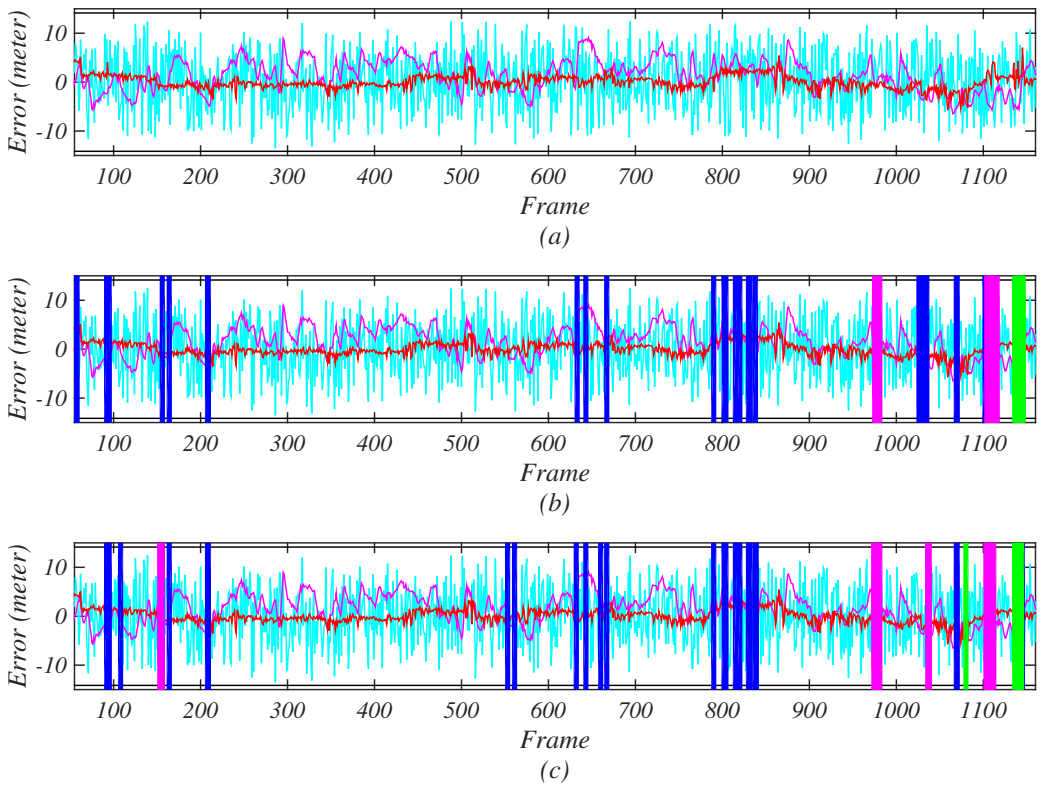


Figure 4.13: *The localization results and detected errors. (a) Localization result of all the frames in a scenario; (b) localization result with ground truth errors; (c) localization result with identified errors.*

Absolute Error (MAE) of the optimized position is $MAE_a = 1.006m$. In Fig. 4.13b, the ground truth errors in localization method are marked on localization result. The blue columns represent localization errors, green ones are lane detection errors, and pink are map errors. These errors lead to unsuccessful vehicle localization. If these error frames are ignored, the MAE decreases to $MAE_b = 0.860m$. Fig. 4.13c is the same localization result, but with the errors identified by error detection models mentioned in Chapter 4. If the error frames are removed, the MAE of lateral vehicle position error is $MAE_c = 0.925m$. MAE_c is greater than the MAE_b, this is because not all the detected errors match ground truth errors. In this experimental scenario, 73 frames are measured as questionable frames but 21 frames of them are not included in ground truth, which means that 52 errors among a total number of 73 are detected correctly, the success ration is 71.23%. Meanwhile, 13 error frames are not identified by the error modeling. These false alarm errors and undetected errors causes a larger MAE in Fig. 4.13c than Fig. 4.13b, which is $MAE_c > MAE_b$. However, MAE_c is lower than MAE_a, which means that the probabilistic error modeling helps to improve vehicle localization performance.

4.5 Conclusion

A probabilistic error modeling is proposed. The confidences of lane detection error model, localization error model and map error model are estimated. The potential errors are decided according to methods based on probabilistic error models. Lane detection errors are determined by both the lane marking confidences and the lane width. Localization errors are modeled according to an error cue that can indicate the fitting quality of MKE based localization. Map topology errors are detected according to the third lane markings from vision and the lane number property from OSM. The error of mismatching between OSM markings and benchmark GPS positions are demonstrated. The results indicate that error modeling improves the vehicle localization performance. Compared to the benchmark, the error modeling algorithm can be modified to further increase the GPS precision of our localization method.

Chapter 5

Global System Analysis

Contents

5.1 Environment Setup

The proposed method was evaluated with real data obtained by an experimental vehicle on highway roads. The route started from Université Paris-Sud 11 to Évry, and returned back to the University region, along Route Départementale such as D128, Route National such as N118, N104 and Autoroute A6. The driving distance is about 75 km, whose track is shown in Fig. 5.1. The whole journey includes straight roads and corner roads, light traffic and heavy traffic, and other common traffic conditions, which are suitable to test and evaluate the feasibility and accuracy of our method. Several representative road stretches of the journey are picked up to verify our algorithms in this chapter. The experiment on this database is regarded as another verification of our algorithm, besides the experimental evaluations of lane detection method (in Section 2.5), vehicle localization method (in Section 3.8) and error modeling algorithm (in Section 4.4) using KITTI public database.

The experimental vehicle is equipped with a monocular pinch hole camera, two low-cost GPS receivers, and a high-cost RTK-GPS receiver. The sensor setup is depicted in Fig. 5.2. The camera is mounted in front and at the central part of the vehicle, capturing the road features ahead of the rover. The camera height is 1.2 meter with respect to the road surface. The three GPSs are located on top of the vehicle roof, the

low-cost GPSs are used to produce rough vehicle positions, while the RTK-GPS one is considered as the ground truth vehicle position for result verifying. The two GPSs are mounted outside the vehicle, to ensure strong connection with satellites. The GPSs are 2 meters behind the camera, this distance is considered in the multi-kernel based vehicle localization.

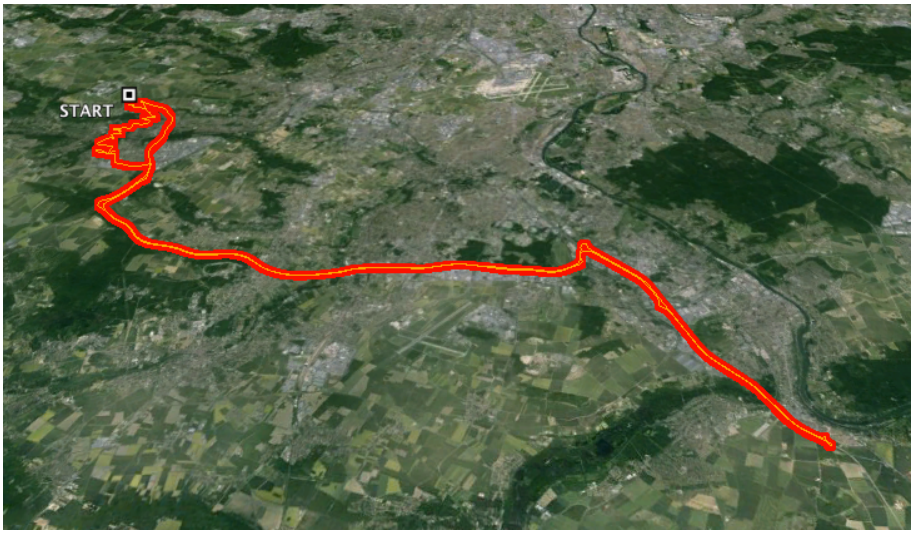


Figure 5.1: *Experimental driving trajectory data on Google map.*

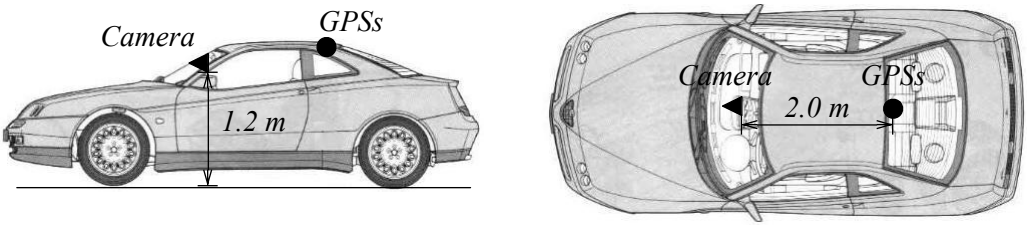


Figure 5.2: *Experimental vehicle equipped with sensors. (a) Vehicle lateral view; (b) vehicle vertical view.*

The camera type is POINT GREY FL2-08S2C-C monocular camera shown in Fig. 5.3a, with the maximum resolution 1032*776, and maximum frame rate 30 FPS, whose wide angle is 45°. In our experiment, the VGA resolution of camera size is set as 640*480, with the frequency of 12.5 Hz, the pitch angle of the lens is 0°.

The low cost GPS in this test is GARMIN GPS 18x LVC, shown as Fig. 5.3b, whose frequency is set up as 1 Hz. The position accuracy under GPS standard positioning service is <15 meter[1].

ALTUS APS-3 GPS receiver, shown as Fig. 5.3c, is selected as the benchmark positioning sensor. The localization accuracy is 1cm in RTK mode, and the maximum output rate is 25 Hz[2]. In our experiment, the set data frequency is 2 Hz.

[1]Garmin GPS technical specifications
[2]Altus GPS technical specifications

UBLOX GPS receiver is depicted as Fig. 5.3d. This GPS receiver uses an LEA-5T as the GPS module, attached with EVK-5T evaluation kit [3]. In the experiment, the frequency of the output data is set up as 2 Hz. However, the data of UBLOX GPS is not included in the experiment, because a large number of the NMEA signals are lost. For instance, in a test scenario of about 30 minutes, an approximated number of 3600 NMEA signals is expected, while only 11.0% (397 signals) from UBLOX are received by RTMAPS platform. The potential reasons of the data loss problem may be: 1) the frequency is too high for the receiver to produce both raw data and NMEA data at the same time; and 2) the embedded Kalman Filter is still enabled.

Figure 5.3: *Camera and GPS receivers in our experiment. (a) POINT GREY camera; (b) GARMIN GPS receiver; (c) ALTUS GPS receiver; (d) UBLOX GPS receiver.*

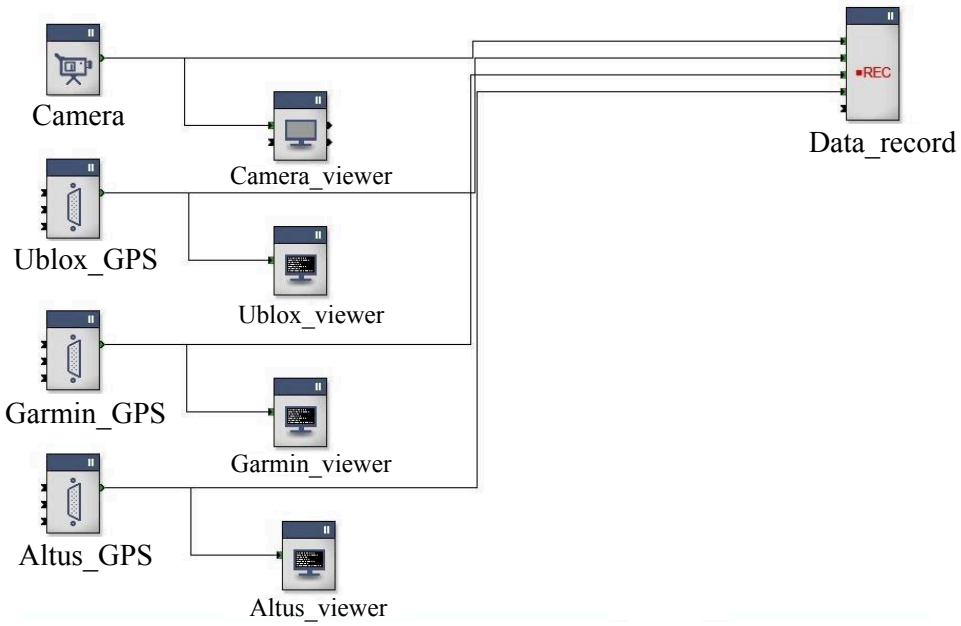


Figure 5.4: *RTMaps diagram of data collection. "Camera", "Ublox_GPS", "Garmin_GPS" and "Altus_GPS" represent sensors; "Camera_viewer", "Ublox_viewer", "Garmin_viewer" and "Altus_viewer" are data monitors, "Data_record" is used to record experimental data.*

To synchronize the vision input, the low-cost GPS data and high-accuracy GPS signal, RTMaps software platform is introduced in our experiment. RTMaps is an asynchronous high performance platform designed to take an advantage of an efficient and

[3] Ublox GPS technical specifications

easy-to-use framework for fast and robust developments. The system diagram used in our test is shown as Fig. 5.4. When configured in RTMaps platform, all the sensors, including camera and GPSs, are represented as components. The "Camera" component means the camera connected with FireWire 1394b interface, while "Ublox_GPS" component, "Garmin_GPS" component and "Altus_GPS" component are the three GPS receivers respectively, connected through USB interface. The data of all the sensors are monitored through "Camera_viewer", "Ublox_viewer", "Garmin_viewer" and "Altus_viewer". The sensor data are recorded by a recording component "Data_record". In the recording component, the camera data are saved as the type of "Video File", and the NMEA format data of GPS are recorded as data stream through the recording method of "stream dump" in RTMaps. Thanks to RTMaps, the data from separated sensors are sealed with uniformed time stamps.

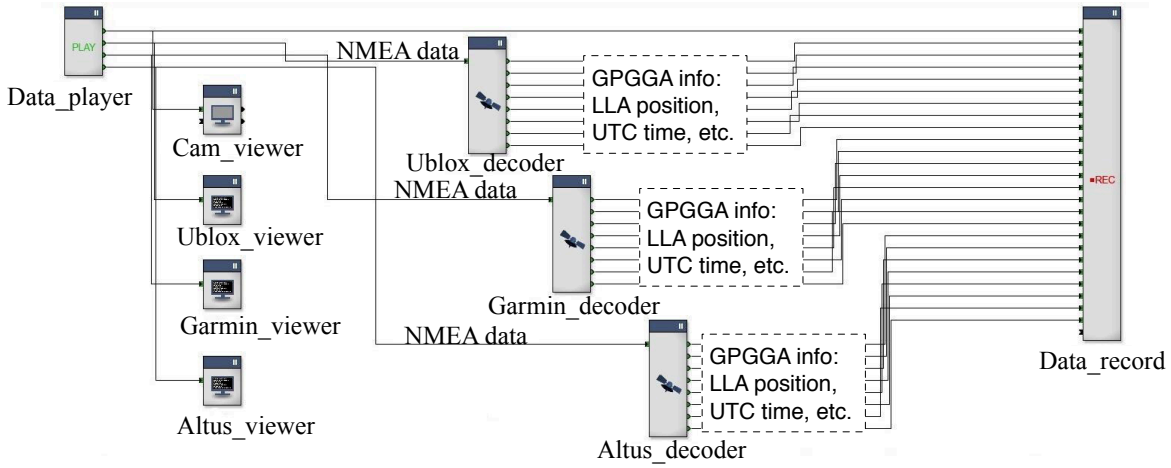


Figure 5.5: *RTMaps diagram of decoding. "Data_player" contains all the raw data; "Cam_viewer", "Ublox_viewer", "Garmin_viewer" and "Altus_viewer" are data monitors; "Ublox_decoder", "Garmin_decoder" and "Altus_decoder" are used to extract the GPGGA information; "Data_record" saves all the derived data.*

To decode the recorded experimental data, the diagram in Fig. 5.5 is utilized. A play component "Data_player" is used to load the data file, whose outputs include camera streams, low-cost GPS signals, and benchmark GPS signals. All the data streams are shown in "Cam_viewer", "Ublox_viewer", "Garmin_viewer" and "Altus_viewer". The camera stream is processed into single images by recording component "Data_record". The GPS signals are not saved directly. The NMEA data from "Data_player" are processed by decoder components "Ublox_decoder", "Garmin_decoder" and "Altus_decoder", to extract GPGGA information. GPGGA information is one track of NMEA data, including the Latitude, Longitude, Altitude (LLA) positions, UTC time and other key data of the the GPS receiver. The decoded GPGGA data of all the GPSs are saved to "Data_record" as "stream dump" file. Besides, a time stamp list of all the images and positions is created at the same time when the sensor data are recorded. The key function of decoding procedure is to extract the useful data in our

algorithm and save these data into proper formats. After the decoding procedure, the image frames, the vehicle positions, and the time stamps of the entire sensor signals are available to our algorithm.

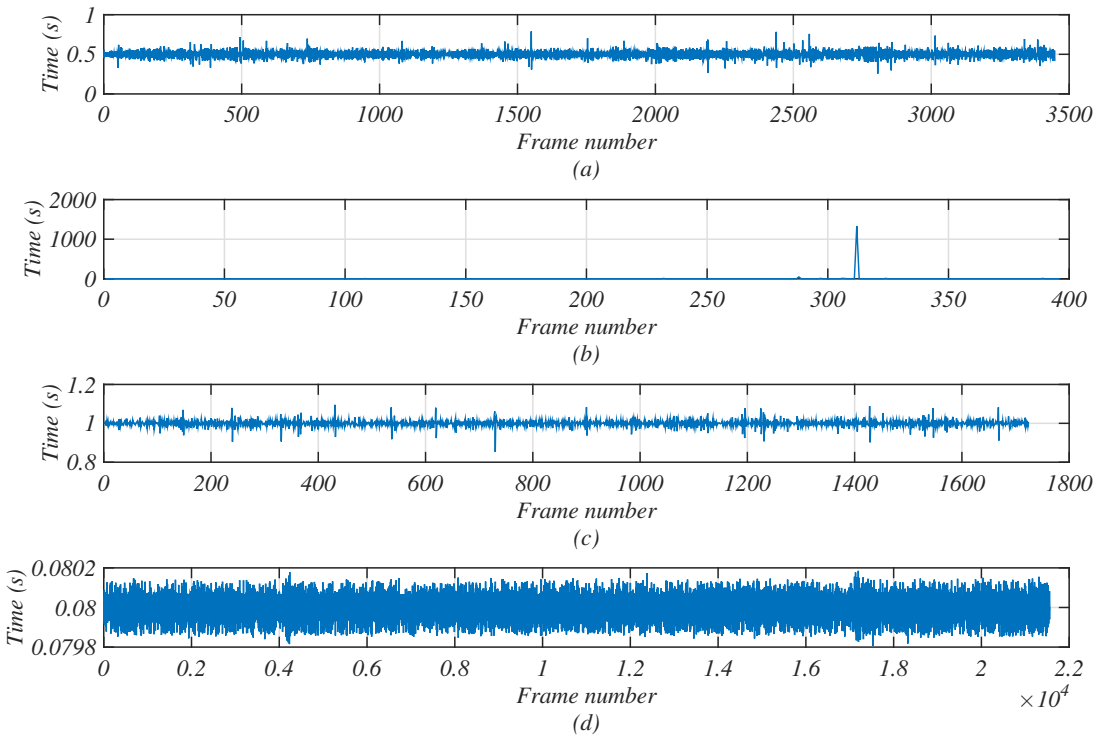


Figure 5.6: *Time cycle tendency of different data source. (a) Output data frequency of ALTUS GPS; (b) output data frequency of UBLOX GPS; (c) output data frequency of GARMIN GPS; (d) output data frequency of camera.*

Till now, the images and the GPS positions are derived, as well as the time stamps. However, these data with different time stamps are not matched because of different frequencies and uneven time cycles, shown as Fig. 5.6. The frequencies of data sources are totally different: camera capture frequency is 12.5 Hz, GARMIN GPS output rate is 1 Hz, and the frequency of ALTUS and UBLOX is 2 Hz. In addition, the time cycle of different GPS outputs are not the same. For instance, the maximum time cycle between two signals in benchmark is 0.7899 second, while the minimum is 0.2556 second. The output time cycle of UBLOX is abnormal, the time cycle increases to 1200 second at frame 320, which implies that about 20 minute data are missing at frame 320. This curve also proves that the UBLOX GPS lost a lot of data during the experiment and only GARMIN GPS is regarded as the low-cost GPS receiver.

In our algorithm, it is assumed that a GPS position is associated to an image, which means a GPS position is required at every image stamp. To this end, the GPS signals, including low-cost GPS (GARMIN) and benchmark GPS (ALTUS, are estimated at each camera time stamp. Take low-cost GPS as an example, a quadratic function on

latitude of time is fitted according to three neighborhood GPS latitude positions. The functions on longitude and altitude are fitted in the same manners. With respect to these two functions of time, the latitude, longitude and altitude of each camera time stamp during these two GPS positions are estimated.

5.2 Goal of Naturalistic Data Experiment

The proposed marking detection and vehicle localization algorithms have been verified using KITTI database. However, implementing an on-vehicle experiment and verifying our methods on the fresh data can be regarded as a further step to evaluate our algorithms. The experiment objectives are listed below:

- At first, the system construction and feasibility can be testified, for instance, multi-source data collection in a high frequency.

- Secondly, the lane marking detection algorithm, including image processing and model fitting, can be tested in a different highway environment other than KITTI.

- Thirdly, the marking based particle filter, the map based lane selection, and marking based vehicle localization are evaluated by more on-road conditions.

- Fourthly, the probabilistic error models can be assessed in a different system.

- Lastly, experiences on the test setup and implementation are accumulated for further improvement.

Two scenarios are selected from our on-vehicle experiment data, nominated as “scenario A” and “scenario B”. Each scenario owns the amount about 1000 image frames. A low-cost GPS coordinate and a benchmark coordinate are attached to each image frame. The chosen scenarios include light traffic, heavy traffic, straight roads, curve roads, multi-lane segments, and other common road environments.

To evaluate our algorithm, benchmark data are created manually frame by frame, for both two scenarios. The benchmark data of lane selection are portrayed according to the camera vision. The benchmark of error modeling are obtained more complicatedly, the camera vision, lane detection results, the map information and localization results are analyzed for each frame. Therefore, the process to obtain error modeling benchmark by hand is time-consuming, this is why two scenarios are selected instead of the whole experimental journey.

5.3 Lane Marking Detection

An example of marking detection result is shown in Fig. 5.7. Fig. 5.7a depicts the perspective image, the blue area represents the IPM area, a vision range of 28 meters in front of the rover is captured. Pink pixels are the "signatures" derived from image processing, while the green pixels are "signatures" with high contribution to marking detections. In the BEV image as Fig. 5.7b, a pair of lane markings is extracted. The BEV image is then processed into binary image according to image processing, shown as Fig. 5.7c. The potential lane marking "signatures" are marked. Finally, two lane markings of the host lane are fitted according to multi-kernel estimation, depicted in Fig. 5.7d. The detected marking are labelled as red pixels and curves in both binary image and perspective image.

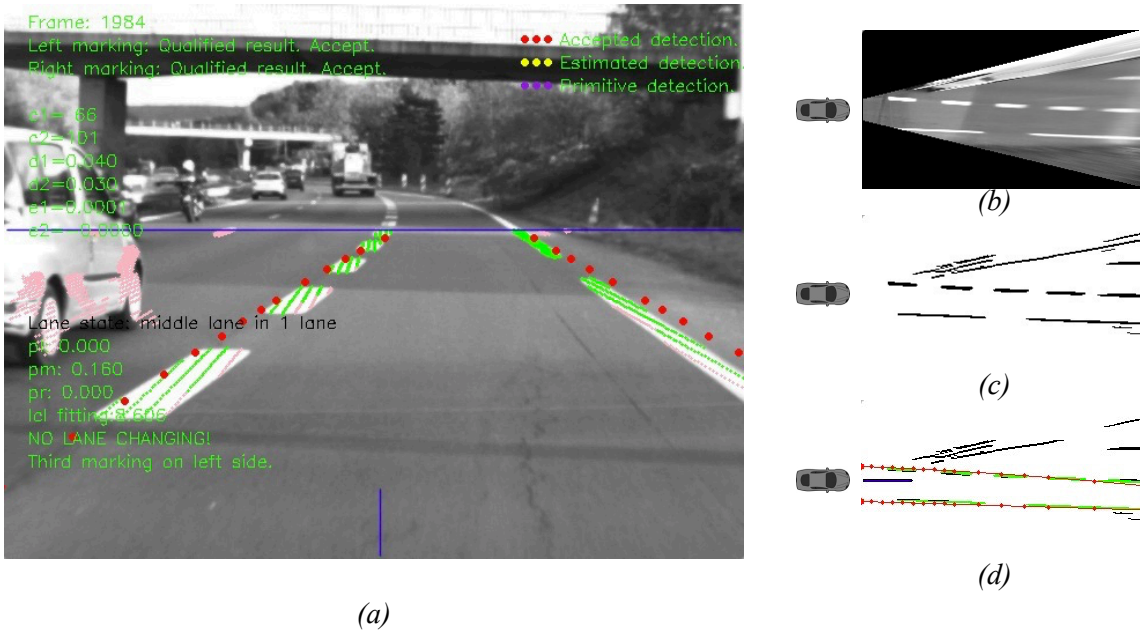


Figure 5.7: *Lane detection example. (a) Perspective view; (b) BEV view; (c) binary view after image processing; (d) detected lane marking in BEV view.*

Different from KITTI database, our experiment data has no uniform benchmark for marking detection. One indicator to evaluate the detection quality is the lane width. The width is represented as $d_{lane} = |c_r - c_l|$, where c_l and c_r are the zero order components of left and right marking models. On a normal highway road, the lane width is between 3.3 meter and 3.6 meter. If the lane width computed from a lane detection result is out of this range, this lane detection is regarded as a potential error detection. Fig. 5.8 is the lane width of scenario A. The range of normal lane width is marked as black dashed lines. In this scenario, 0.89% (89 frames) of the frames own unusual lane width, shown as Fig. 5.8b. In scenario B, shown in Fig. 5.9, the unusual width rate is 1.27% (127 frames).

In Fig. 5.9a, a lane width of 4 meter is derived at frame 5440. This is caused according

to a road branch. A 4-lane road is divided into two 2-lane roads at this area, handing to different directions. At frame 5440, our algorithm detected the right side marking of the current road, but the left side marking of the other branch road, leading to a wrong marking detection.

In Fig. 5.8a and Fig. 5.9a, the lane width varies "discretely". The width change between two frames is a multiple of 0.1m, because in BEV images, one pixel represents 0.1m in reality.

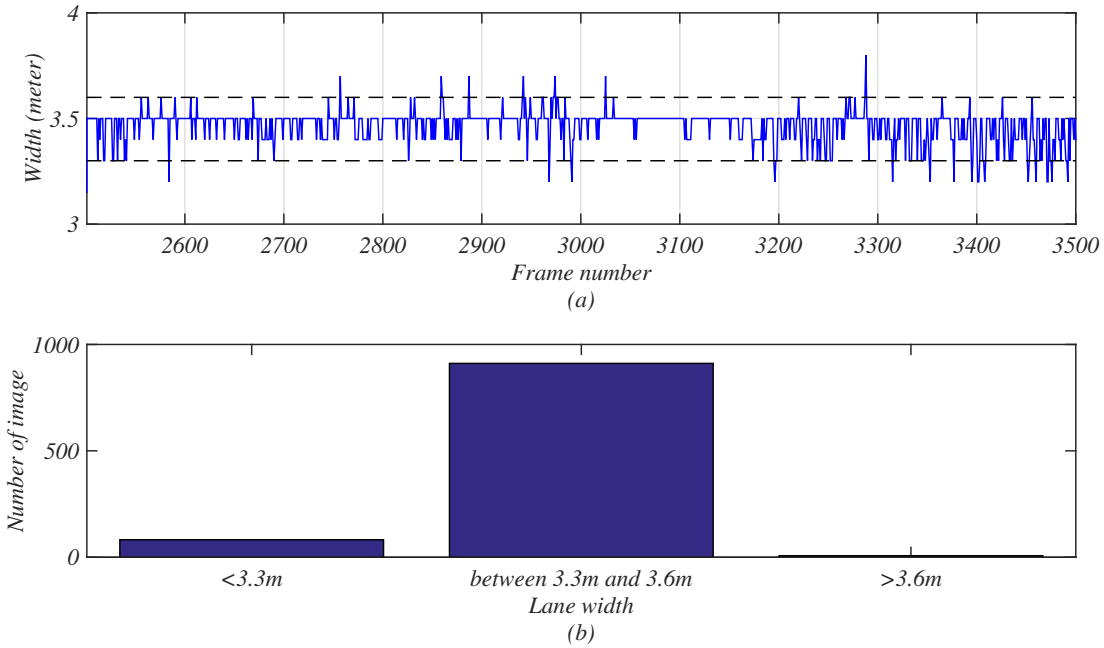


Figure 5.8: *Lane width of scenario A. (a) Lane width variation; (b) width histogram.*

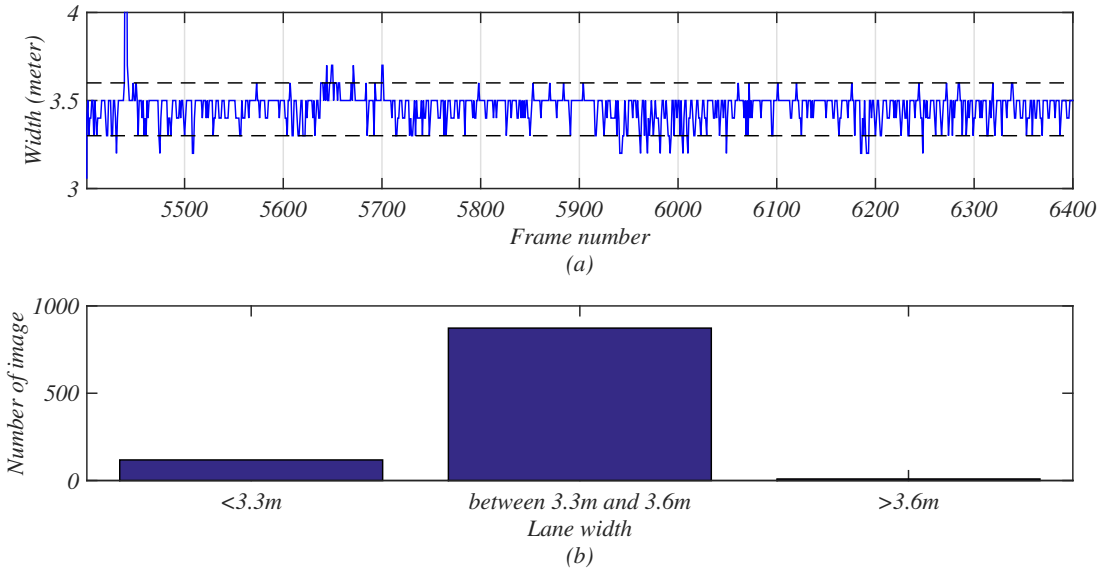


Figure 5.9: *Lane width of scenario B. (a) Lane width variation; (b) width histogram.*

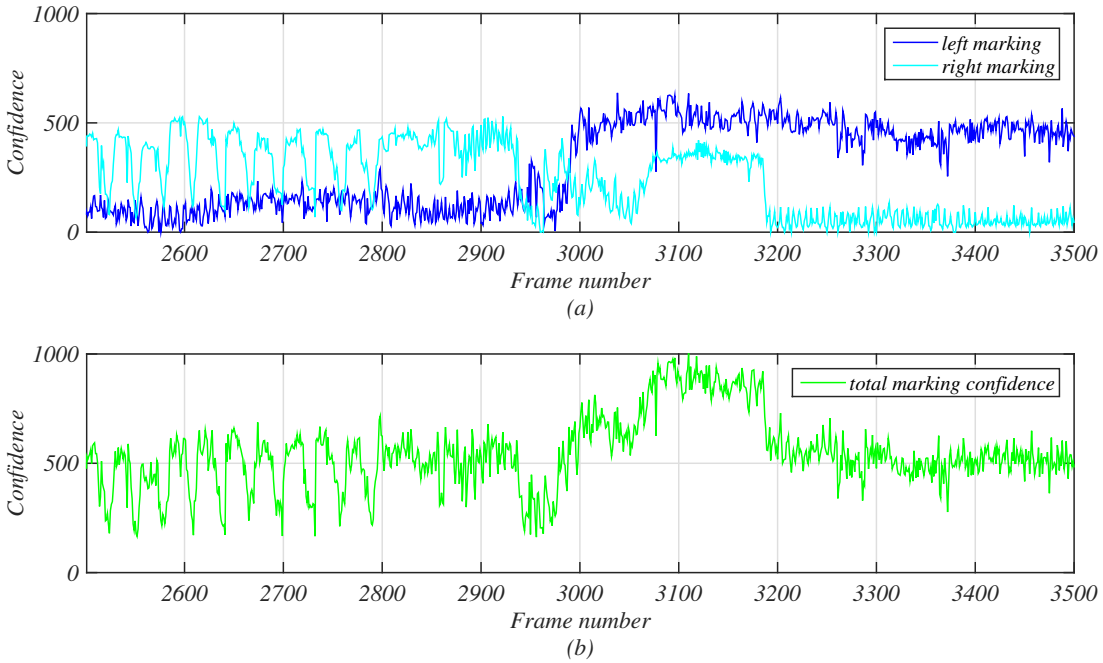


Figure 5.10: *Marking confidence of scenario A. (a) Single marking confidence; (b) total marking confidence.*

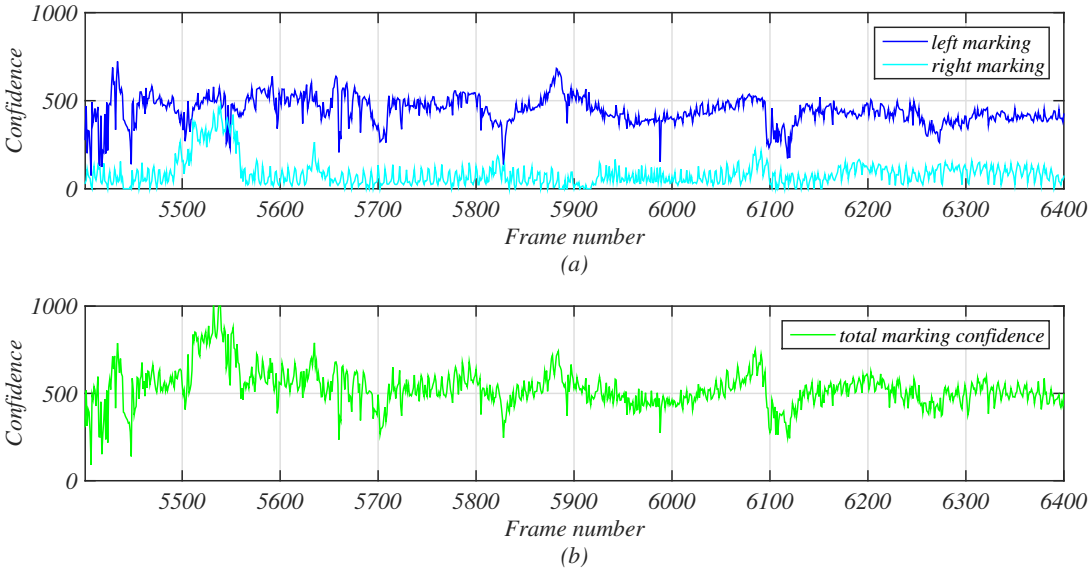


Figure 5.11: *Marking confidence of scenario B. (a) Single marking confidence; (b) total marking confidence.*

Another indicator to verify the results is the detection confidence $conf_{ld}$ detailed in Chapter 2. Fig. 5.10 and Fig. 5.11 give the confidence of the lane markings of the two scenarios. In Fig. 5.10a, blue curve and cyan curve are the confidence of left and right markings respectively. From frame 2500 to frame 2950, right side has higher confidence than left side, while from frame 3200 to 3500, the left side is higher instead. In Fig.

5.11a, the confidences of left side are greater than the right side in most of the frames. The reason is that in highway situations, one side of the marking can be dashed lines or blurry markings, whose confidence is therefore lower.

In our algorithm, the sums of marking confidence on both sides are considered together to avoid false judgement with respect to low confidence of one side. The sum confidence is depicted as green curves in Fig. 5.10b and Fig. 5.11b. The green curve has less fluctuation than blue and cyan curves, which means that the sum confidence is more stable than single confidence, and is able to reflect the marking quality better. A threshold of sum confidence is used to classify qualification markings. As an example, value the confidence as 200, only 13 of 1000 frames have a false detection in scenario A. The confidence curve in scenario B is shown in Fig. 5.11. 7 frames have a confidence lower than the threshold among 1000 frames.

Fig. 5.12 and Fig. 5.13 show the time cost of lane marking detection algorithm. In our algorithm, two states are selected taking into account of the lane detection quality. Tracking state is adopted if the previous detection owns a confidence higher than the confidence threshold, otherwise, detection state is used. The state selection is presented in Section 2.4.

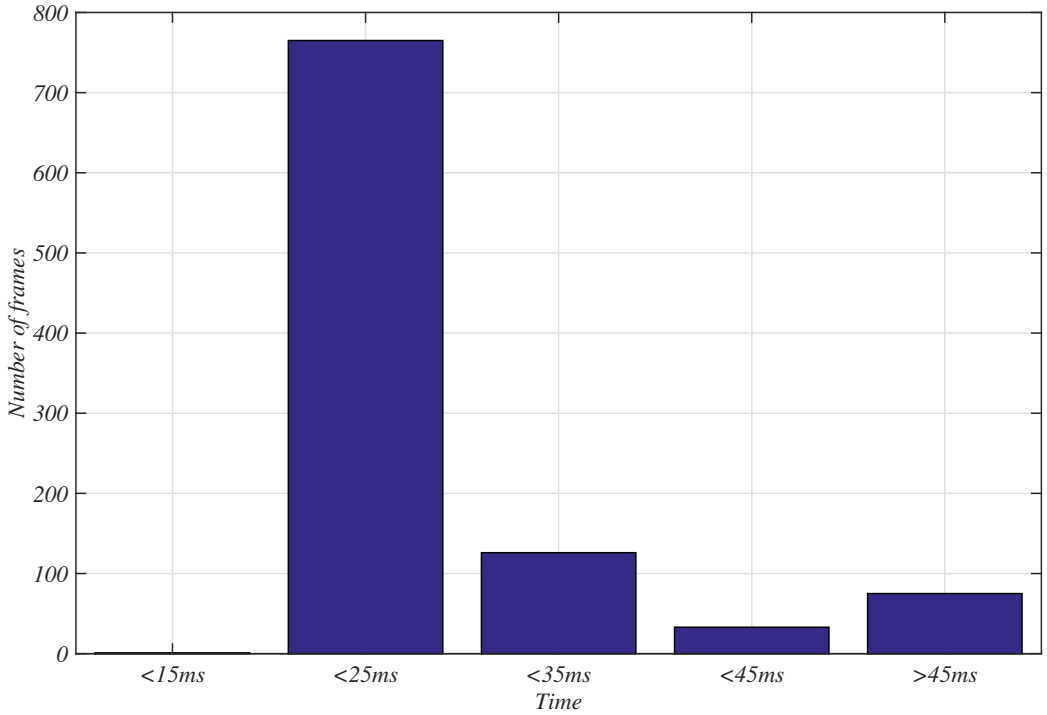


Figure 5.12: *Time cost of scenario A.*

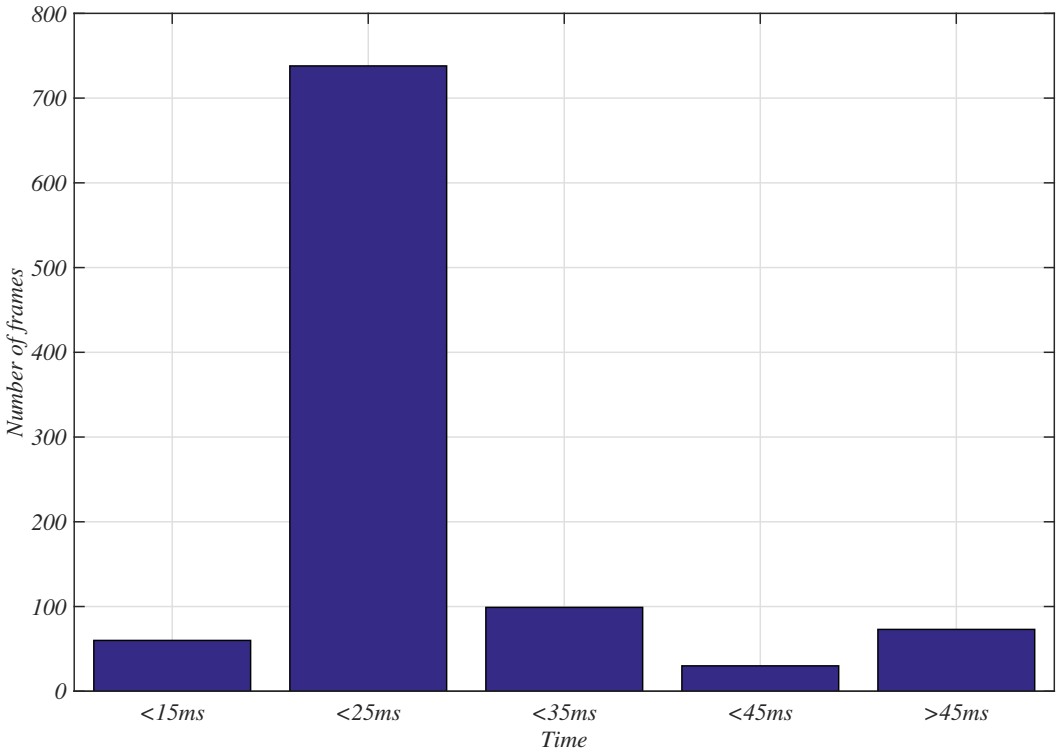


Figure 5.13: *Time cost of scenario B.*

The average time of scenario A is 23.1 ms, while the average time of detection states is 47.2 ms, and that of tracking state is 20.2 ms. In scenario B shown in Fig. 5.13, the detection state frames owns a mean value of 46.8 ms, on the other side, the mean of tracking states is 19.0 ms. The total average of these two scenarios are 21.1 ms. This value is similar as the time cost using KITTI database detailed in Section 2.5.

5.4 Vehicle Localization

The first step in vehicle localization is lane selection stage, which determines which lane the vehicle is in within multi-lane road sections. Fig. 5.14 and Fig. 5.15 show the result of lane selection in scenario A. In our algorithm, 6 different states are considered in a maximum 3-lane road section, which are 1 lane, right lane of 2-lane, left lane of 2-lane, right lane of 3-lane, middle lane of 3-lane and left lane of 3-lane. These states are depicted on the y-axis of Fig. 5.14. The original state, represented as blue lines, is the lane section state by the original GPS positions. Red lines are the filtered lane states after our lane selection stage. The green lines are the benchmark in this scenario, which are defined manually. The accuracy of original state is 54.1%, while the accuracy of our algorithm is 87.7% in this scenario, the lane states are corrected from original state to filtered state in more than 230 frames. In scenario B, the accuracy rate is 75.7%.

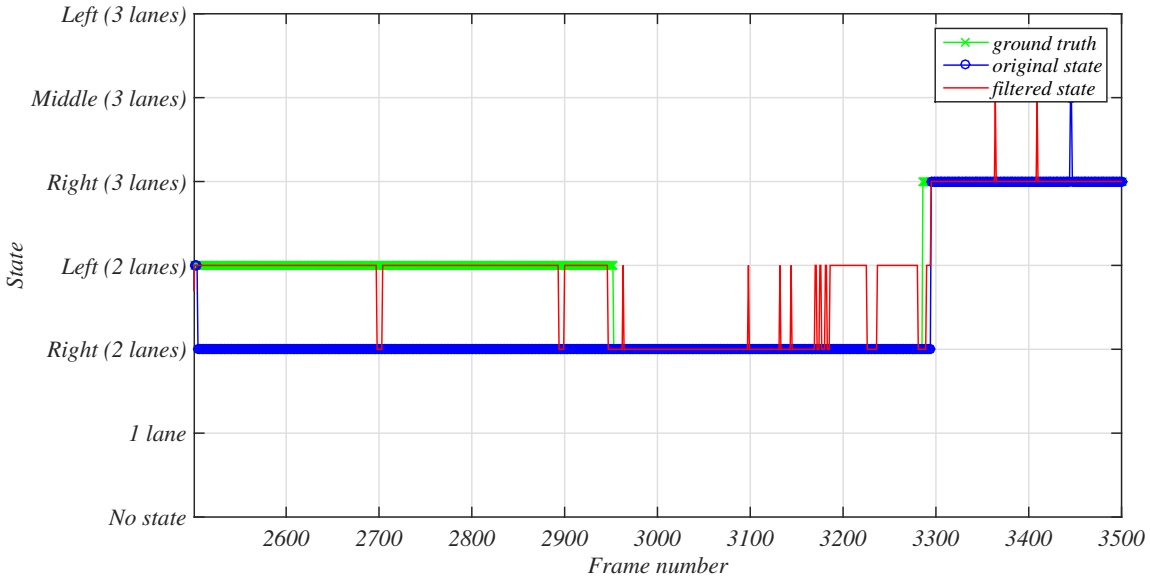


Figure 5.14: *Lane selection of scenario A.*

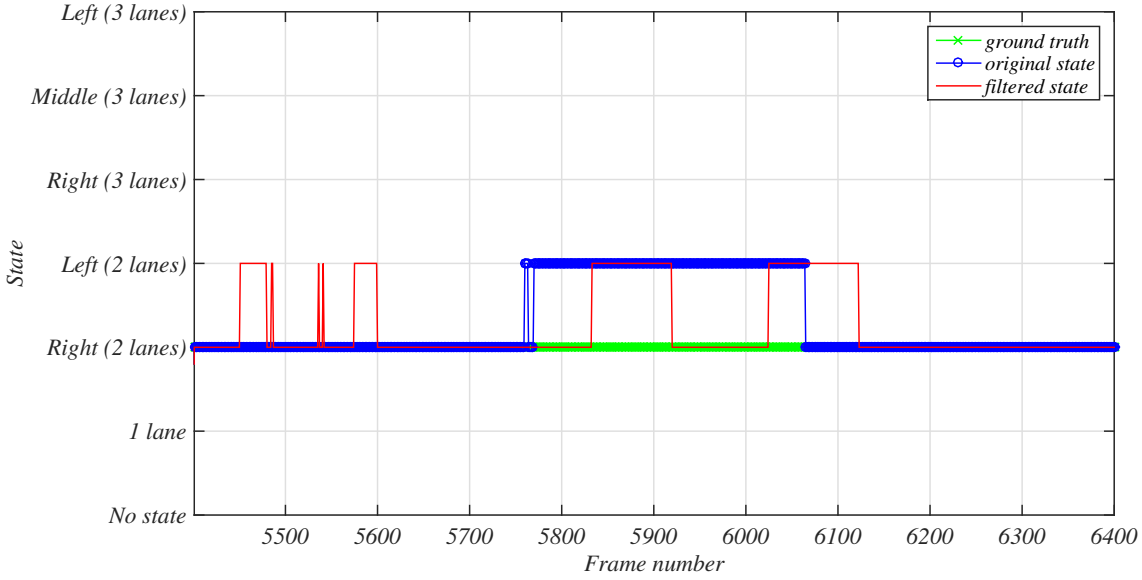


Figure 5.15: *Lane selection of scenario B.*

However, our lane selection algorithm is affected by noises. For instance, at about frame 3200 to 3300 in Fig. 5.14, a road barrier is detected as the third lane marking on the right side. The specific detections of a frame during frame 3200 to 3300 are depicted in Fig. 5.16. The lane markings of current road lane are detected as red pixels in Fig. 5.16a, the potential third lane markings on left and right sides are searched. A third marking on the lefthand side is obtained, as the pink pixels in the red block of Fig. 5.16a. A clearer view is shown as BEV space in Fig. 5.16c. The "signatures" of road barriers cause this left lane marking. So the current state is determined as the left lane by mistake. Improving the image processing and adopting a camera with wider

open angle can be possible solutions to avoid this error detection.

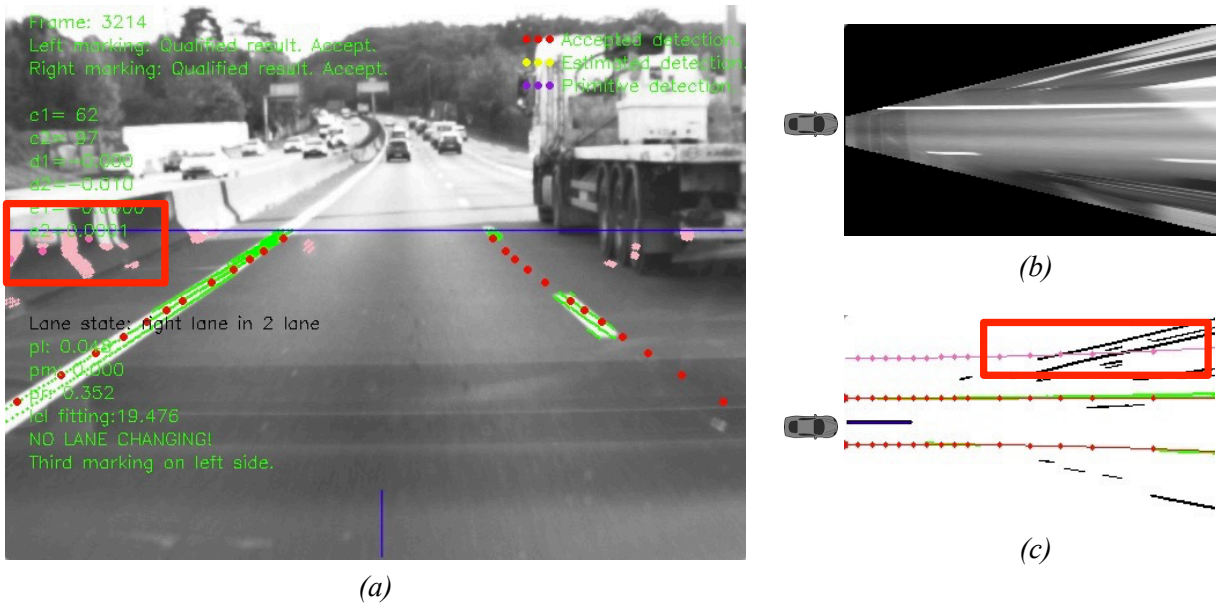


Figure 5.16: *A false example of lane selection affected by noises. (a) Perspective view; (b) BEV view; (c) detected lane marking in BEV view.*

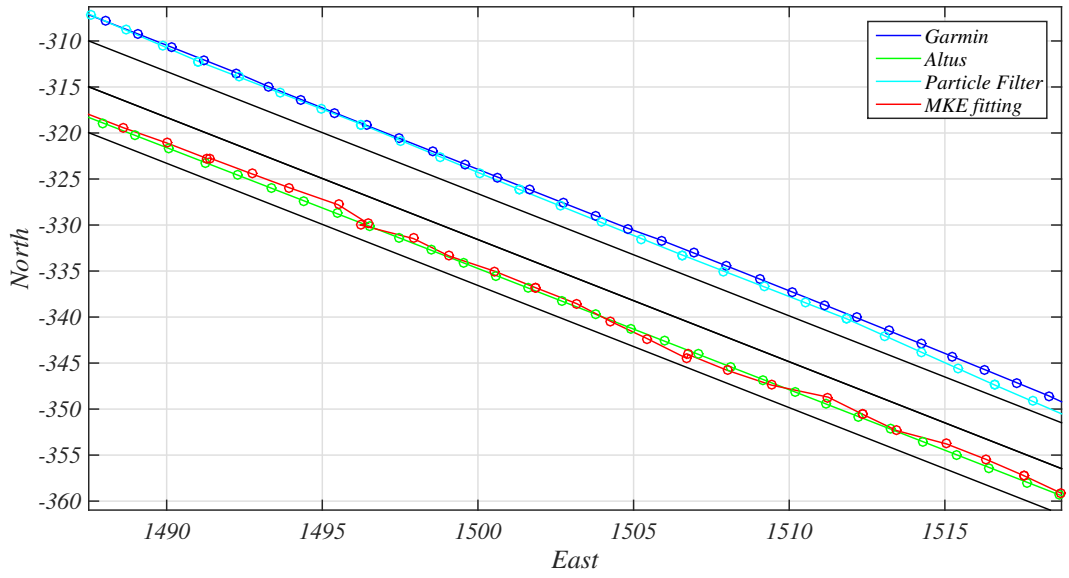


Figure 5.17: *An example scene of localization result in scenario A.*

Fig. 5.17 shows an example of the localization result in ENU space. Black lines are the lane markings derived from OpenStreetMap database, the blue dots are the position coordinates provided by low-cost GPS, cyan ones are the positions processed by marking based particle filter, the red dots are the localization result produced by the Multi-kernel estimation, the green ones are the benchmark. In this example, the vehicle is driving on the right lane of a 2-lane road according to the trajectory of the benchmark curves (green ones). However, the track of low-cost GPS (blue) is at the edge

the left lane, even out of the road area. These original positions are at first adjusted by particle filter, using the direction of lane markings. The multi-kernel method is implemented on the filtered position derived from particle filter, the vehicle trajectory is corrected again to the host lane, shown as the red lines in Fig. 5.17. The red curve is nearer to the benchmark, and even overlapping the benchmark curve on the vehicle lateral direction. This example scene implies that our algorithm can adjust the rough GPS positions to the ground truth locations.

The error curve of the localization result in scenario A is shown in Fig. 5.18. Blue curve is the error of rough GPS position, the filter position using particle filter is depicted as cyan, and the localization result using MKE method is the red curve. In 95.6% of the frames, the localization result has smaller error than rough coordinates. A localization result with more than 3 meter's error is caused by several reasons. For instance, at frames around 2700 and frames around 2950, a bad lane marking is detected because of the heavy traffic, which affects the MKE localization. During these wrong lane detections, the marking confidences support the results, because a number of traffic vehicle bodies are regarded as lane marking "signatures" in image processing. At frame 3200 to 3300, a false lane state is produced in the lane selection stage, the vision based lane markings are fitted with the lane marking of a wrong road lane, so the localization errors increase. At frame 3300 to 3500, the MKE based vehicle localization method works without problems, but the error at this part is around 2 meter. This is caused by inaccurate road location of OSM, which is detailed in Section 4.2.4. This problem, which is able to produce errors in method evaluation, cannot be detected in our error modeling algorithm.

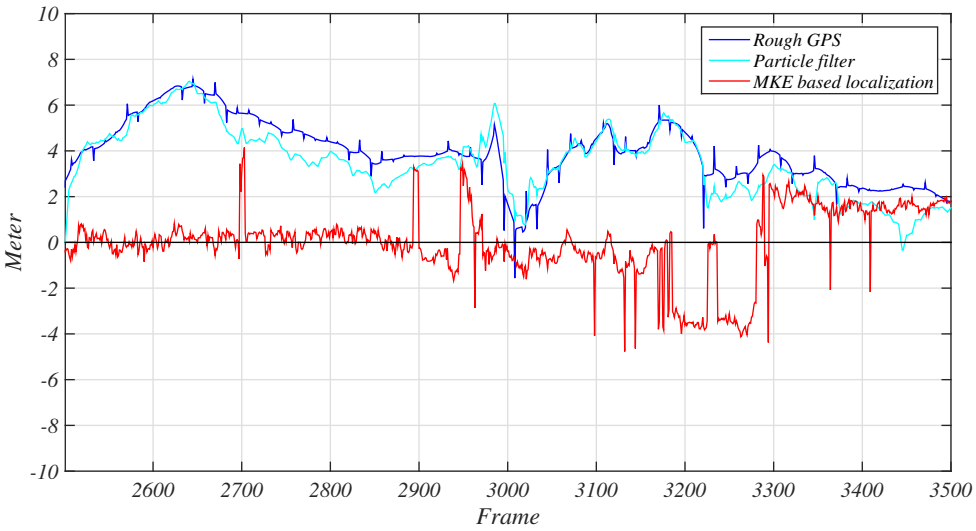


Figure 5.18: *Errors of localization result in scenario A.*

The localization error of scenario B is depicted in Fig. 5.19. Four road parts own larger error than other parts, which are road parts from frame 5450 to frame 5500, from frame

5550 to frame 5650, from frame 5850 to frame 5950, and from frame 6050 to frame 6150. The errors of these four parts are caused by wrong lane selection results. Notice that these four parts match the false lane selection results in Fig. 5.15.

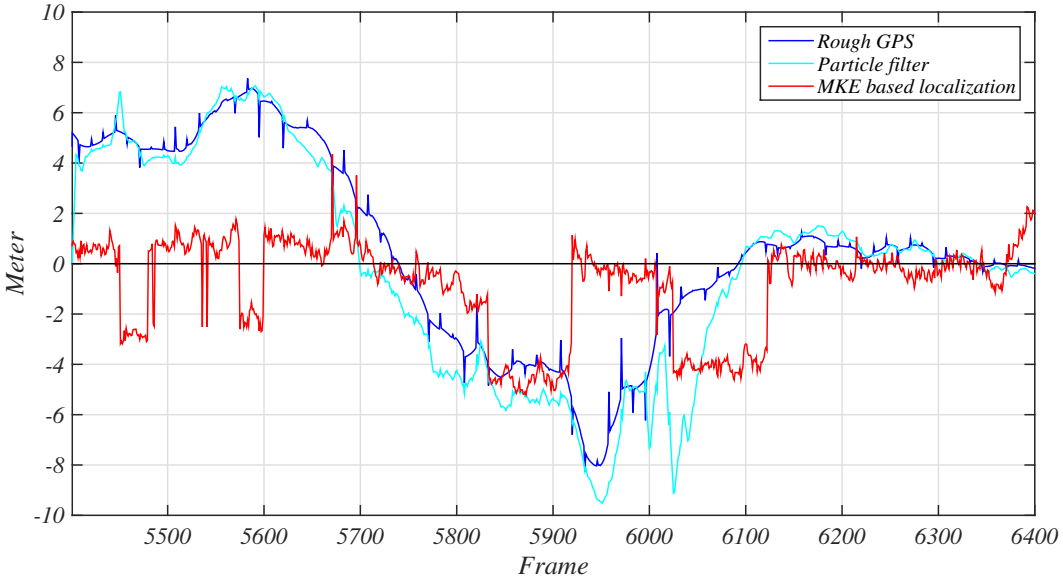


Figure 5.19: *Errors of localization result in scenario B.*

The statistics of the two scenarios are listed in Table 5.1 and Table 5.2. In Table 5.1, both the average and the MAE of particle filter are less than the original positions, which implies that the errors are alleviated by marking based particle filter. The effect of particle filter is not very obvious because the track of low cost GPS is relative smooth. The average error decreased from 3.96 meter to −0.0054 meter, the MAE is reduced from 3.96 meter to 1.10 meter after the MKE localization. The errors are greatly reduced by our algorithm, which represents that our localization method is able to adjust the rough vehicle position, using open source database and camera vision.

Table 5.1: *Error statistics of scenario A.*

	Low-cost GPS	Particle filter localization	MKE localization
average error	3.96m	3.53m	-0.00543m
standard deviation	1.36m	1.52m	1.50m
max value	7.13m	7.04m	7.04m
95th percentile	6.47m	6.14m	3.51m
MAE	3.96m	3.53m	1.10m

Table 5.2: *Error statistics of scenario B.*

	Low-cost GPS	Particle filter localization	MKE localization
average error	0.536m	-0.172m	-0.848m
standard deviation	3.72m	4.24m	1.87m
max value	7.37m	7.07m	7.07m
95th percentile	6.65m	7.24m	4.52m
MAE	2.95m	3.41m	1.40m

In Table 5.1, The MAE decreases from 2.96 meter to 1.40 meter shows the improvement of localization. But the average error of rough GPS is less than the localized position, because between frame 5000 to frame 5750, the average error of rough GPS is positive, and turns to negative during frame 5750 to frame 6100. In this scenario, the statistics of particle filter localization are greater than the rough GPS input. The vehicle dynamics used in our marking based particle filter, including speed and vehicle heading, are estimated through rough GPS positions. So the estimated vehicle dynamics are not as precise as the ground truth, and thus lead to errors in particle filter.

Consequently, in Table 5.1 and Table 5.2, the MAE of MKE localization is less than the low-cost GPS positions, which numerically prove that the marking based vehicle localization helps to improve the performance of vehicle fix.

5.5 Error Modeling

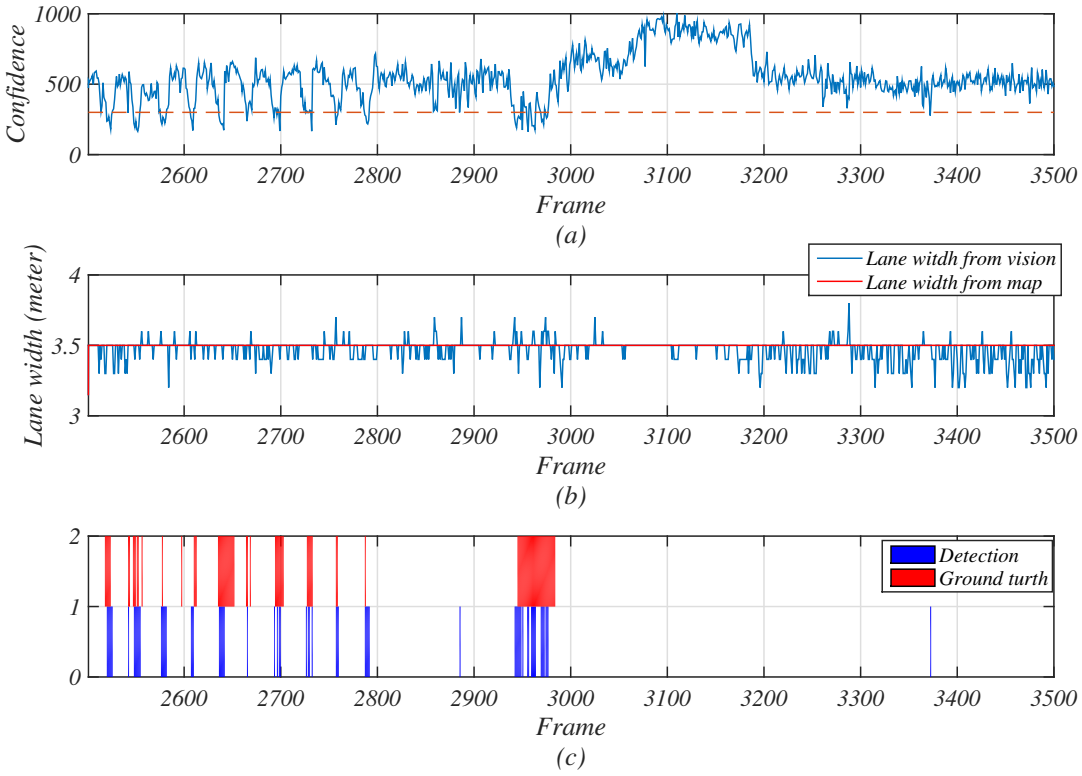


Figure 5.20: *Lane detection error of scenario A. (a) Lane detection confidence, blue curve is the confidence curve, red curve is the threshold; (b) lane width curve; (c) detected error.*

Error modeling is implemented to locate the potential problematic localization results and to improve the localization in a further step. The first error model is the lane detection error. Fig. 5.20 and Fig. 5.21 depict the cues and the result of the lane detection error. The first cue is the marking confidence, shown as Fig. 5.20a and Fig. 5.21a, confidence lower than the threshold is a prerequisite of lane detection error. The

second cue is the lane width from both vision source and map source, depicted as Fig. 5.20b and Fig. 5.21b. A large width difference can be a potential error. Herein the width difference is assigned as 20 cm as an example. When a low confidence and an unusual width difference happen in one frame, a lane detection error is inferred, shown as Fig. 5.20c and Fig. 5.21c. The blue lines in Fig. 5.20c and Fig. 5.21c represent detected errors, the red lines are the error benchmark as a comparison.

Among the benchmark lane detection errors of scenario A, 46.9% (45/96) of the errors are detected. The unsuccessful detections come from two sources, FP (False Positive) and FN (False Negative). For instance, around frame 2780, the lane markings are dashed road signs, whose amount of "signatures" is very limited, a low confidence is provided here. In addition, the zero order component estimation is affected by noises in a dash lane marking, so the lane width can be imprecise. Therefore, although the detected marking model from lane detection algorithm is correct, the cues implies a bad detection, an FP occurs. An FN scene happens around frame 2650, a number of marking "signatures" are produced by the heavy traffic, a qualified marking result is evaluated by the algorithm, and thus no error is discovered.

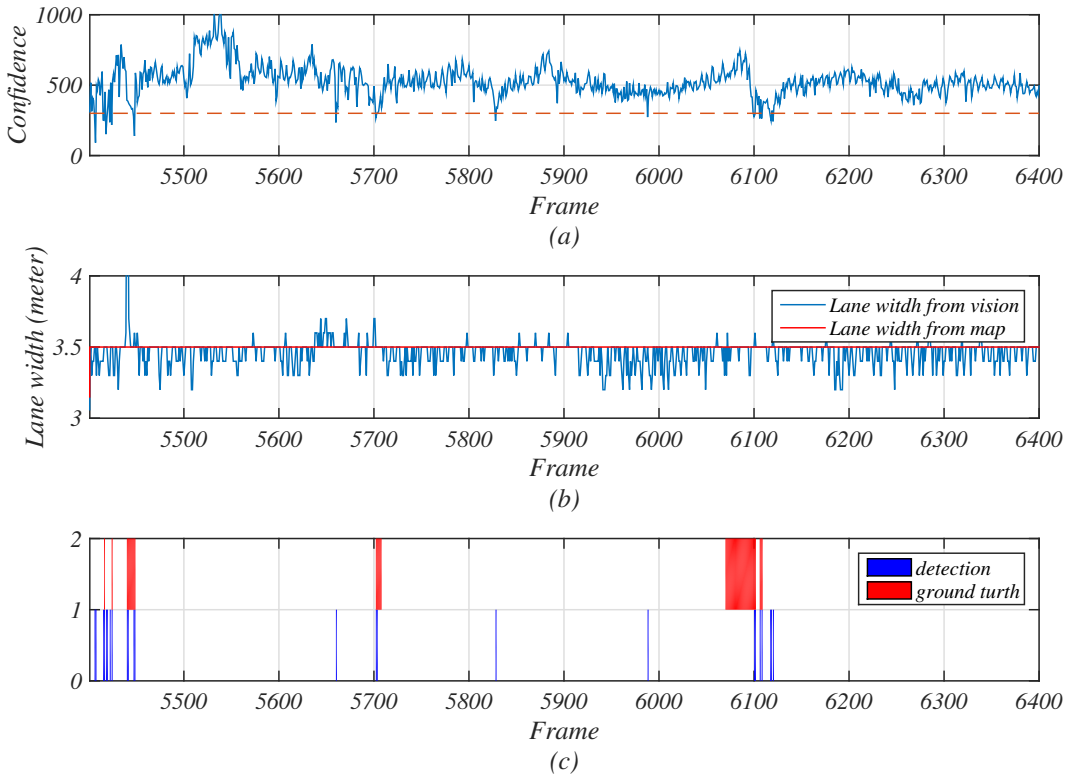


Figure 5.21: *Lane detection error of scenario B. (a) Lane detection confidence, blue curve is the confidence curve, red curve is the threshold; (b) lane width curve; (c) detected error.*

In scenario B, 44.23% (23/52) lane detection errors are inferred. During frame 6050 to frame 6100, FN situations happen because of heavy traffic. The bodies of other

vehicles can produce noise lane marking “signatures”, which affect the lane marking confidence.

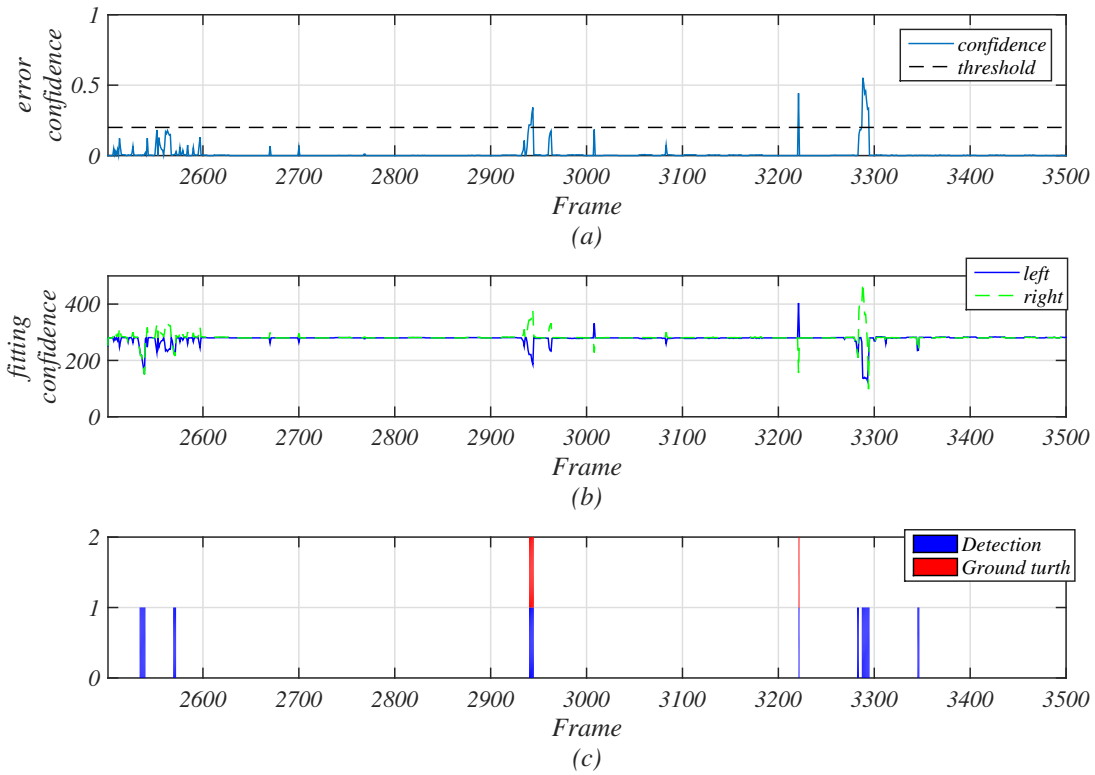


Figure 5.22: *Localization error of scenario A. (a) Localization confidence; (b) left and right side fitting confidence; (c) detected errors.*

The error of localization is shown in Fig. 5.22 and Fig. 5.23. The fitting confidences of both left and right marking in MKE localization stage are depicted in Fig. 5.22b and Fig. 5.23b. The error confidence of localization error model is computed according to the left and right side fitting confidence, shown as Fig. 5.22a and Fig. 5.23a. The frames with errors are listed in Fig. 5.22c and Fig. 5.23c, as well as the ground truth. In scenario A, all the 6 frames with localization error are detected. However, 21 False Positive (FP) detections are produced. When the markings from vision and map are fitted, the pixels within the BEV image (about 28 meter in front) are considered. And in our algorithm, the markings from map include only the markings of the current lane, and the markings of one next possible lane cell. In this scenario, the lengths of some lane cells are less than 10 meters in BEV space image, which means that the fitting confidence is not adequate, and therefore leads to FP detections. In scenario B, 74.1%(20/27) of localization error frames are detected. The FP conditions are caused by the same reason analyzed in scenario A.

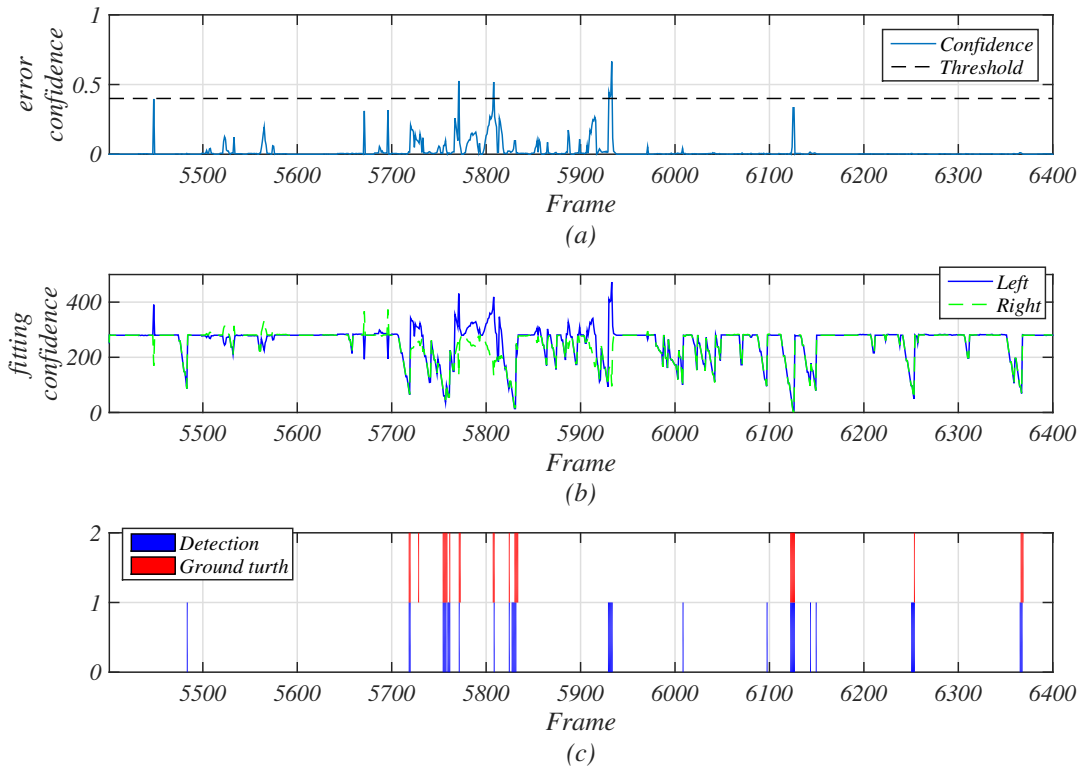


Figure 5.23: *Localization error of scenario B.*

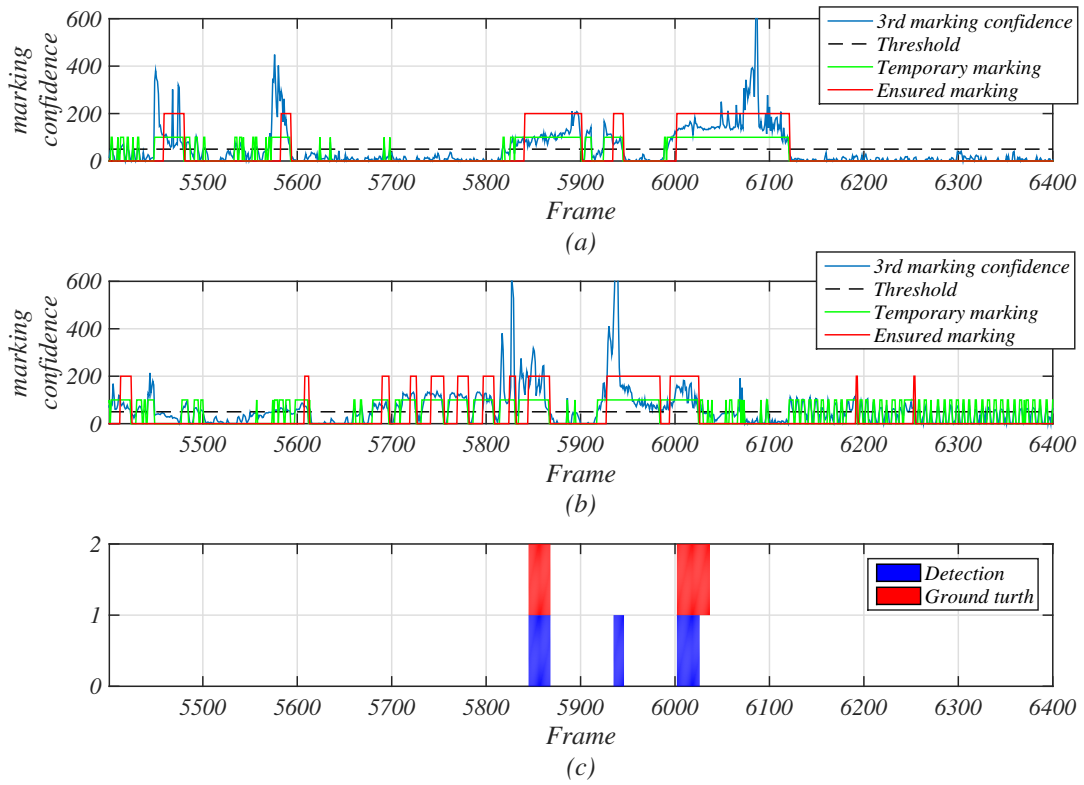


Figure 5.24: *Map topology error of scenario B. (a) Left third marking confidence; (b) right third marking confidence; (c) detected errors.*

The map topology error only appears in scenario B, shown as Fig. 5.24. In Fig. 5.24a, the confidence of left side third marking is shown as blue curve. The confidence threshold is marked as a black dashed line, the threshold is set to 50 as an example. The confidences which are greater than the threshold are determined as temporary third markings, depicted as green curves, while the stable third markings of the left side are marked as red curves. The third markings on the right side are detected in the same manners shown as Fig. 5.24b. The detected errors and benchmark errors are illustrated in Fig. 5.24c. At frame 6000, both left and right third lane markings are detected, but the rover is in a 2-lane road from map database. A logical error occurs, and a map topology error is located. Among the 58 benchmark map topology errors, 47 frames are inferred by our algorithm.

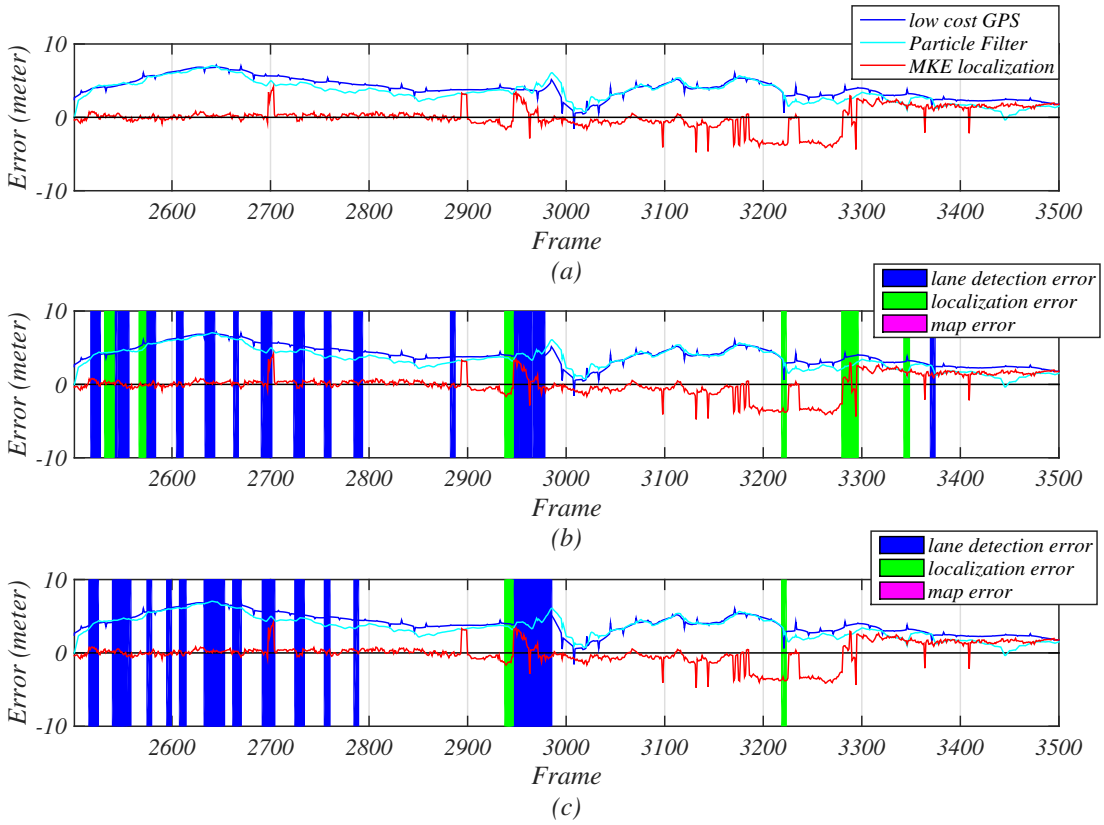


Figure 5.25: *The localization results and detected errors of scenario A. (a) MKE localization result; (b) MKE localization results with detected error models; (c) MKE localization results with benchmark error models.*

Finally, the error overview is shown in Fig. 5.25 and Fig. 5.26. In scenario A, Fig. 5.25a shows the vehicle localization result obtained in Section 5.4. The three error models are introduced into the localization result in Fig. 5.25b. Blue lines are the lane detection error models, green lines are the localization errors, pink blocks represent map topology errors. Some frames with large lateral error are detected by the error modeling. For instance, around frame 2950, several lane detection errors are detected,

these frames sequences own a large error because of lane detection. And around frame 3300, the localization errors are detected, matching the frames with large lateral error caused by the MKE fitting. The benchmark is shown in Fig. 5.25c. When the frames with errors are removed, the MAE of lateral error has been decreased from 1.10 meter to 0.98 meter, which implies that the localization performance is improved using error modeling. Furthermore, the MAE of lateral error with benchmark error models is 0.87 meter, which means that the accuracy of localization can be further improved if the error modeling is more precise.

Fig. 5.26 illustrates the error model detections of scenario B. The MAE with error modeling decreases from 1.40 meter to 1.26 meter, comparing to the localization result, while the MAE of the benchmark error model is 1.11 meter. The corresponding indices, including the vehicle localization error, the localization error with detected error modeling and the localization error with benchmark error modeling, decrease successively. These data support the effect of error modeling, as well as the potential improvement.

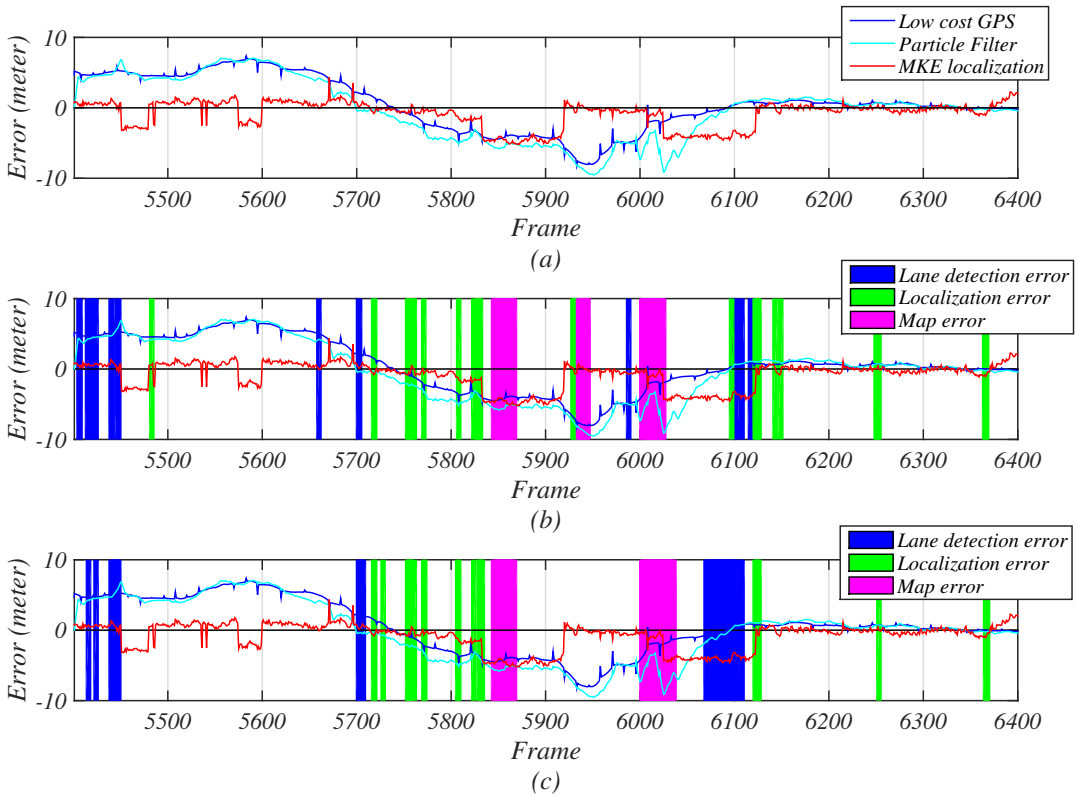


Figure 5.26: *The localization results and detected errors of scenario B. (a) MKE localization result; (b) MKE localization results with detected error models; (c) MKE localization results with benchmark error models.*

5.6 Experiment Analysis

The experiments using our vehicle test data verify the vehicle test platform, the lane marking detection method, vehicle localization algorithm, and error modeling approach.

The vehicle test platform is composed of a camera, GPS sensors, and RTMaps software. The data from camera and GPSs are recorded and synchronized in RTMaps. The experimental results prove the feasibility and efficiency of this platform. However, the data from UBLOX GPS receiver are not useful to evaluate our algorithm because of problematic setup. In addition, the performance of lane selection and error modeling are restricted by the current camera open angle.

In the lane detection part, the lane markings are detected in real-time. The markings on both straight roads and corner roads are fitted. Although the lane detection algorithm is not assessed directly by ground truth markings, the evaluation according to lane width and the results of marking based particle filter and MKE localization have proved the marking detection result. However, some improvements are required in this algorithm. The image processing stage imports noises in heavy traffic conditions. The marking confidence provides few contributions to reveal these noises. The marking confidence produces FN results in situations where both markings are dashed lines.

The localization algorithm has reduced the lateral error from 3.96 meter to 1.10 meter. The error modeling has further improved the localization result. In the experiment using KITTI database, it is implied that the rough GPS errors, the errors adjusted by marking based particle filter, and the errors after MKE localization decrease successively. In other words, the lane marking based particle filter is regarded as a first guess of localization, and MKE based localization further improves the vehicle positioning. However, in this real vehicle test, the effect of particle filter is not as obvious as in test using KITTI database. In scenario A, the MAE decreases 0.4m from rough GPS to particle filter. However, in scenario B, the MAE increases 0.4m. The potential reasons are: 1) the estimation of vehicle velocity and heading in real test are not accurate enough to support particle filter; 2) the parameters of particle filter (i.e. number of particle, threshold in resampling) are not well managed.

In error modeling part, the lane detection errors, the localization errors and the map topology errors are detected. The results show that the precision of vehicle localization with error modeling has improved, compared to the vehicle localization with MKE method. However, FP and FN detections still exist in the error modeling method, which limit the output performance.

The results of on-vehicle experiment match the results using KITTI database.

The MAE of position error has been corrected from 3.96m to 0.98m in scenario A using our algorithm, while from 2.95m to 1.26m in scenario B. These results show that our

marking based vehicle localization method effectively improves the vehicle positioning on real vehicle test.

Conclusions and Outlook

Conclusions

Vehicle localization is the primary information needed for advanced tasks like vehicle navigation. This information is usually provided by the use of Global Positioning System (GPS) receivers. However, the low accuracy of GPS in urban environments makes it unreliable for further treatments. The combination of GPS data and additional sensors can improve the localization precision.

Accordingly, we propose a marking feature based vehicle localization using low-cost GPS, monocular camera and open source map. Our method makes use of multi-criterion confidences to infer potential errors, and in advance, to enhance the vehicle localization. At first, the vision-based lane marking models are obtained. Meanwhile, the map-based lane markings of current state are derived from map databases. Both lane marking sources are fused together to implement vehicle localization, using a multi-kernel based algorithm. In order to further improve the localization performance, a probabilistic error model is employed to identify the possible errors.

In lane marking detection method, the perspective image is captured from camera vision. The image is first projected into BEV space, a second-order derivative filter and a cell-based blob algorithm are implemented to extract the lane marking "signatures", and thus binarize the BEV image. The parabola lane marking models are fitted in two steps. The zero order components of left and right lane markings are estimated in multi-cue initialization. This step provides a first guess of model fitting, and reduces the time cost in model parameter fitting. Then the lane markings are obtained according to Multi-Kernel Estimation (MKE) method. To evaluate the quality of marking detection, a self-assessment indicator is introduced, as well as the indicator threshold. The markings with high confidences (higher than threshold) are determined as successful detection.

In marking based vehicle localization, the lane markings from two different sources are compared and fitted to obtain vehicle position fix. At first, the database from OpenStreetMap (OSM) are analyzed and reconstructed to form "lane marking topology" map data. The basic unit in this "marking topology" is a "cell", which is a rectangular with the information of lane markings. Then, a lane marking based particle filter is

exploited to offer a first correction of vehicle localization. The marking direction helps to improve the rough GPS positions. After that, the current lane is derived by lane selection algorithm. The "cell" in OSM, which the rover is in, is estimated according to third marking cue, lane changing cue, and history data cue. On one side, when the current cell is obtained, the lane markings from "marking topology" map data can be derived. On the other side, the vision based lane markings are detected in lane detection algorithm. Both two marking sources are projected to BEV space, and MKE method is implemented to derive the adjusted vehicle fix.

To further improve the vehicle localization performance, a probabilistic error modeling is exploited. Error model of lane detection is the error happened in lane detection algorithm, the marking confidence and lane width are used to locate this error. Error model of localization represents the error occurs in MKE localization, which can be detected by the fitting confidences of left and right side markings. Error model of map topology implies the topology errors in OSM. The third lane markings, and the OSM properties can detect an error of this kind.

The algorithm has been tested on two databases, KITTI public database, and our own on-vehicle experiment. The results show that the lane markings are detected in a high success ratio, and a clear improvement in localization accuracy is achieved by our method.

Future Research

In lane detection part, our algorithm is tested mainly on KITTI database, some difficult situations are not considered. For instance, the methods to avoid noised caused by heavy traffic have to be added. Vehicle detection can be a possible solution to this problem. Meanwhile, the marking confidence should have the ability to distinguish lane marking "signatures" and "noises". The pixels color and a weighted strategy can help on this improvement.

In map reconstruction, the lane markings at road junctions are not accurate, because only the road sections on the vehicle driving route are considered. All the road branches have to be dealt with to perfect the marking reconfiguration at road junctions.

In the marking based particle filter, the inaccurate estimation of vehicle dynamics restricts the localization performance. Vision based odometry may be an alternative to predict the vehicle velocity and vehicle heading.

In lane selection part, wrong results of current chosen cell affect strongly the localization performance. Normally, an extra error of 3.5m (a distance of lane width) is created in a false lane selection. The vehicle trajectory prediction can be an additional cue to determine the current road cell. Meanwhile, the lane changing ability and the third marking success ratio are required to improve.

In MKE localization method, the same parameter setup in lane marking fitting is adopted. However, the usages of MKE method are different. Therefore, different parameter assignment in localization can improve the localization result.

In error models, the FP and FN detections have to be alleviated. The method is to increase the success ratio of cue confidences. These confidences related strongly to previous steps. For instance, if the lane marking confidence can be improved, the effect of lane detection error will be better. More error cues can be introduced into the algorithm. One idea is to use a second journey data to address lane detection error and localization error.

In the on-vehicle experiment, a camera with wider open angle is required, because more vision of third lane marking can be captured. This improvement can benefit the lane selection and error modeling. In addition, the GPS setup should be verified before the experiment, ensuring all the GPS data are useful.

Appendix A

Coordinates Mapping

A.1 Introduction

Three coordinates spaces are referred in the proposed marking based localization method: the perspective space, the Bird's Eye View (BEV) space, and the East-North-Up (ENU) space. We can observe a general view of these coordinates in Fig. A.1. Fig. A.1a depicts the on-vehicle camera setup, blue areas are the whole vision captured by the on-vehicle camera. The obtained camera picture in perspective space is illustrated in Fig. A.1b. We can also observe BEV space and ENU space in Fig. A.1c.

Perspective space $\mathbb{P}^2$ is a 2D coordinate system located lying on the image plane that defines the vision perception work-space of a perspective camera. It is an intersected plane derived from a 3D world space through the center point of projection. Coordinates of a point in the image plane in $\mathbb{P}^2$ are defined as ${}^{PER}\mathbf{p} = [x_{per},\, y_{per}]^T$.

BEV space here is defined as the elevated view of the road environment above the rover. A 3D BEV space is represented as $\mathbb{B}^3$, whose metric is 0.1 meter per pixel. A pixel in $\mathbb{B}^3$ is given as ${}^{BEV}\mathbf{p} = [x_{bev},\, y_{bev},\, z_{bev}]^T$. Fig. A.1c observes the projection of $\mathbb{B}^3$ on $x_{bev}Oz_{bev}$ plane, which is defined as $\mathbb{B}^2$, with a point coordinate ${}^{BEV}\bar{\mathbf{p}} = [x_{bev},\, z_{bev}]^T$.

The local ENU coordinate is organized using "East", "North", "Up" as coordinate axis, whose unit is meter. Normally, a 3D ENU space is represented as $\mathbb{R}^3$, whose measurement is denoted as ${}^{ENU}\mathbf{p} = [x_{enu},\, y_{enu},\, z_{enu}]^T$, where x_{enu}, y_{enu}, and z_{enu} are East, North, and Up direction coordinates respectively. Assume that the road surface is flat, only x_{enu} and y_{enu} are considered in our research. We can define $\mathbb{R}^2$ with the point coordinate ${}^{ENU}\bar{\mathbf{p}} = [x_{enu},\, y_{enu}]^T$ to represent this assumption. The relationship of BEV space and ENU space is given in Fig. A.1c. The distinct differences of BEV space and ENU space are measurement metric and coordinate direction. In this work, ENU coordinates are derived from World Geodetic System 1984 (WGS-84) standard datum.

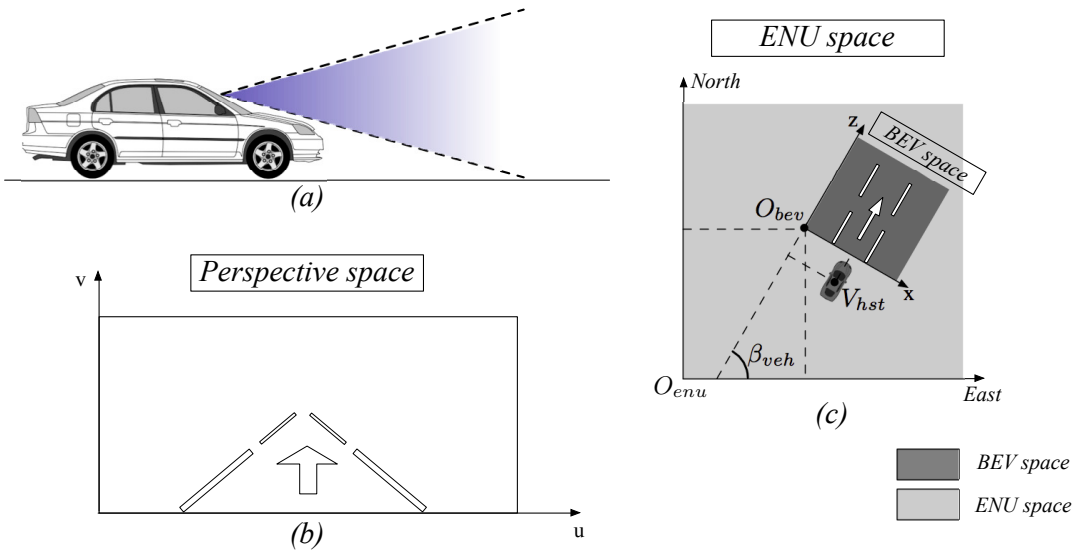


Figure A.1: *Camera implementation and coordinates definition. (a) Camera setup; (b) perspective image from camera; (c) ENU space $\mathbb{R}^2$ (light gray) and BEV space $\mathbb{B}^2$ (dark gray).*

A.2 Mapping between Perspective Space and BEV Space

Inverse Perspective Mapping (IPM) defines the mapping between a random point ${}^{PER}\mathbf{p}$ in image plane (perspective space) and its corresponding point ${}^{BEV}\mathbf{p}$ in BEV space Hartley and Zisserman (2003). The transformation function of IPM relates to camera setup. Fig. A.2 depicts $\mathbb{B}^3$ according to the rover's current state. z_{bev} points to the vehicle's driving direction. In the set, the rover center point is not the coordinate origin, because this coordinate set makes sure the majority of the camera image is projected to the first quadrant of $x_{bev}Oz_{bev}$ plane, which brings convenience in image processing. Denote the rover center position ${}^{BEV}\mathbf{p_{veh}} \in \mathbb{B}^3$, the coordinate of camera center point is represented as ${}^{BEV}\mathbf{p_{cam}} = {}^{BEV}\ \mathbf{p_{veh}} + [h_{cam,x},\ h_{cam,y},\ h_{cam,z}]^T$, where $[h_{cam,x}, h_{cam,y}, h_{cam,z}]^T$ represents the mounted camera location with respect to vehicle position ${}^{BEV}\mathbf{p_{veh}}$. α_{cam}, β_{cam} and γ_{cam} represent pitch angle, yaw angle and roll angle of the camera rotating along x_{bev}, y_{bev} and z_{bev} axis respectively. These parameters are camera extrinsic parameters, representing the camera installation with respect to BEV frame. When extrinsic parameters are known, a point in BEV space ${}^{BEV}\mathbf{p} \in \mathbb{B}^3$ can be projected onto the image plane ${}^{PER}\mathbf{p} \in \mathbb{P}^2$. The procedure is expressed as follows:

$$\begin{bmatrix} u_{per} \\ v_{per} \\ w_{per} \end{bmatrix} = \mathbf{K} \cdot \mathbf{R_X} \cdot \mathbf{R_Y} \cdot \mathbf{R_Z} \cdot \mathbf{T_{XYZ}} \cdot \begin{bmatrix} x_{bev} \\ y_{bev} \\ z_{bev} \\ 1 \end{bmatrix}. \tag{A.1}$$

where

$$\mathbf{K} = \begin{bmatrix} f_{u,cam} & 0 & u_{o,cam} & 0 \\ 0 & f_{v,cam} & v_{o,cam} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{A.2}$$

is camera internal parameter matrix, $f_{u,cam}$ and $f_{v,cam}$ are the camera focal length, $[u_{cam,o}, v_{cam,o}]^T$ is focal point;

$$\mathbf{R_X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{cam} & -\sin\alpha_{cam} & 0 \\ 0 & \sin\alpha_{cam} & \cos\alpha_{cam} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A.3}$$

$$\mathbf{R_Y} = \begin{bmatrix} \cos\beta_{cam} & 0 & -\sin\beta_{cam} & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta_{cam} & 0 & \cos\beta_{cam} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A.4}$$

$$\mathbf{R_Z} = \begin{bmatrix} \cos\gamma_{cam} & -\sin\gamma_{cam} & 0 & 0 \\ \sin\gamma_{cam} & \cos\gamma_{cam} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A.5}$$

are the rotation matrices according to camera angle α_{cam}, β_{cam} and γ_{cam};

$$\mathbf{T_{XYZ}} = \begin{bmatrix} 1 & 0 & 0 & h_{cam,x} \\ 0 & 1 & 0 & h_{cam,y} \\ 0 & 0 & 1 & h_{cam,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A.6}$$

is camera translation matrix, determined by $h_{cam,x}$, $h_{cam,y}$ and $h_{cam,z}$. Eq. A.1 can also be expressed as

$$\begin{bmatrix} u_{per} \\ v_{per} \\ w_{per} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} x_{bev} \\ y_{bev} \\ z_{bev} \\ 1 \end{bmatrix}. \tag{A.7}$$

Assuming $y_{bev} = 0$, Eq. A.7 can be simplified to

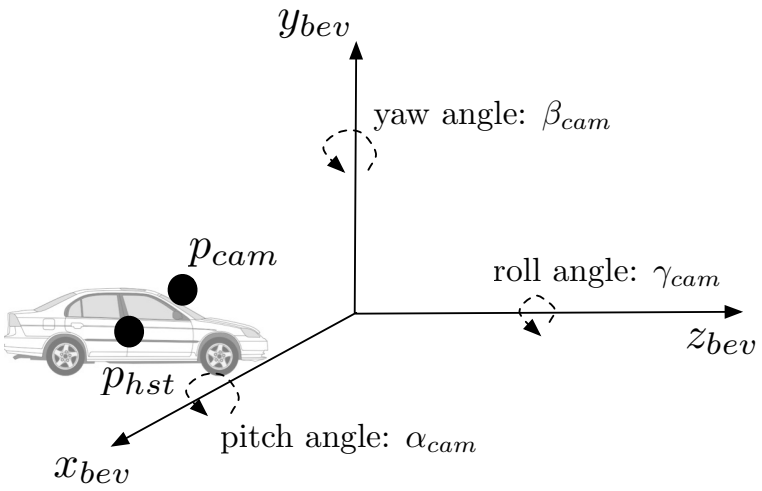


Figure A.2: *Camera setup in BEV space.*

$$\begin{bmatrix} u_{per} \\ v_{per} \\ w_{per} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{13} & m_{14} \\ m_{21} & m_{23} & m_{24} \\ m_{31} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} x_{bev} \\ z_{bev} \\ 1 \end{bmatrix}. \tag{A.8}$$

And point ${}^{PER}\mathbf{p}$ in perspective space is derived as follows

$$x_{per} = \frac{u_{per}}{w_{per}} = \frac{m_{11} \cdot x_{bev} + m_{13} \cdot z_{bev} + m_{14}}{m_{31} \cdot x_{bev} + m_{33} \cdot z_{bev} + m_{34}}, \tag{A.9}$$

$$y_{per} = \frac{v_{per}}{w_{per}} = \frac{m_{21} \cdot x_{bev} + m_{23} \cdot z_{bev} + m_{24}}{m_{31} \cdot x_{bev} + m_{33} \cdot z_{bev} + m_{34}}. \tag{A.10}$$

Eq. A.9 and Eq. A.10 can be represented as the mapping function from $\mathbb{B}^2$ to $\mathbb{P}^2$ with the following equation:

$${}^{PER}\mathbf{p} = f_{IPM}^{-1}\left({}^{BEV}\bar{\mathbf{p}}\right). \tag{A.11}$$

Under the assumption that all the transformation matrices are non-singular, the mapping from perspective frame to BEV space can be simplified as

$${}^{BEV}\bar{\mathbf{p}} = f_{IPM}\left({}^{PER}\mathbf{p}\right). \tag{A.12}$$

A.3 Mapping between BEV Space and ENU Space

A reference point is selected as the origin of local ENU space $\mathbb{R}^2$. Assuming that the road surface in camera range is a plane (i.e. ground flatness assumption), only 2D coordinates $\mathbb{R}^2$ and $\mathbb{B}^2$ are considered. The transformation between BEV space and ENU space is illustrated as Fig. A.1c where O_{bev} and O_{enu} stand for the origins of BEV coordinate and ENU coordinate respectively, V_{hst} represents the rover position, β_{veh} is the yaw angle of the rover. In order to transform to ENU coordinate, the

BEV coordinate needs to rotate, zoom, and translate on North and East coordinate. According to the motions of BEV coordinate mentioned above, the expression from BEV space to ENU space is defined by Eq. A.13.

$$\begin{bmatrix} x_{enu} \\ y_{enu} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\sin\beta_{veh}}{T_{b2e}} & \frac{\cos\beta_{veh}}{T_{b2e}} & x_{enu}^{O_{bev}} \\ -\frac{\cos\beta_{veh}}{T_{b2e}} & \frac{\sin\beta_{veh}}{T_{b2e}} & y_{enu}^{O_{bev}} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{bev} \\ y_{bev} \\ 1 \end{bmatrix}, \tag{A.13}$$

where $T_{b2e} = 0.1$ is a metric coefficient, used to meet different metrics in $\mathbb{R}^2$ and $\mathbb{B}^2$ and

$$^{ENU}\mathbf{O_{bev}} = \left(x_{veh}^{enu} - \frac{x_{veh}^{bev}}{T_{b2e}} \cdot \sin\beta_{veh} - \frac{y_{veh}^{bev}}{T_{b2e}} \cdot \cos\beta_{veh}, y_{veh}^{enu} + \frac{x_{veh}^{bev}}{T_{b2e}} \cos\beta_{veh} - \frac{y_{veh}^{bev}}{T_{b2e}} \sin\beta_{veh} \right), \tag{A.14}$$

are the coordinates of O_{bev} in ENU space, $^{BEV}\bar{\mathbf{p}}_{\mathbf{veh}} = \left[x_{veh}^{bev}, y_{veh}^{bev}\right]^T$ and $^{ENU}\bar{\mathbf{p}}_{\mathbf{veh}} = [x_{veh}^{enu}, y_{veh}^{enu}]^T$ are the coordinates of $\bar{\mathbf{p}}_{\mathbf{veh}}$ in $\mathbb{B}^2$ and $\mathbb{R}^2$ space respectively. The rigid transformation from ENU to BEV is given as:

$$\begin{bmatrix} x_{bev} \\ y_{bev} \\ 1 \end{bmatrix} = \tag{A.15}$$

$$\begin{bmatrix} T_{b2e} \cdot \sin\beta_{veh} & -T_{b2e} \cdot \cos\beta_{veh} & T_{b2e} \cdot \left(-x_{enu}^{O_{bev}} \cdot \sin\beta_{veh} + y_{enu}^{O_{bev}} \cdot \cos\beta_{veh}\right) \\ T_{b2e} \cdot \cos\beta_{veh} & T_{b2e} \cdot \sin\beta_{veh} & -T_{b2e} \cdot \left(x_{enu}^{O_{bev}} \cdot \cos\beta_{veh} + y_{enu}^{O_{bev}} \cdot \sin\beta_{veh}\right) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{enu} \\ y_{enu} \\ 1 \end{bmatrix}.$$

Eq. A.13 and Eq. A.15 can be simplified as:

$$^{ENU}\bar{\mathbf{p}} = f_{ENU}\left(^{BEV}\bar{\mathbf{p}}\right), \tag{A.16}$$

$$^{BEV}\bar{\mathbf{p}} = f_{ENU}^{-1}\left(^{ENU}\bar{\mathbf{p}}\right). \tag{A.17}$$

Thus, the mapping functions mentioned in Fig. A.3 are introduced. A point in one the three spaces can be projected to another space using these four mapping functions: f_{IPM}, f_{IPM}^{-1}, f_{ENU} and f_{ENU}^{-1}.

A.4 Mapping from WGS-84 to ENU Space

GPS position is an essential part to finalize vehicle localization. To simplify calculation, it is necessary to convert WGS-84 standard datum into ENU space coordinate Grewal

et al. (2007). WGS comprises a standard coordinate system for the Earth, latitude, longitude, and height. This datum conversion is accompanied through Earth-Centered-Earth-Fixed (ECEF) frame, which is an intermediate frame. At first, a geodetic position ${}^{WGS}\mathbf{p} = [p_{lat},\, p_{lon},\, p_{alt}]^T$, which denotes the GPS measurement of latitude, longitude and altitude, can be converted to ECEF coordinate ${}^{ECEF}\mathbf{p} = [x_{ecef},\, y_{ecef},\, z_{ecef}]^T$ using the following formula:

$$\begin{aligned} x_{ecef} &= \left(\frac{a}{\sqrt{1 - e^2 \cdot \sin^2 p_{lat}}} + p_{alt} \right) \cdot \cos p_{lat} \cdot \cos p_{lon}, \\ y_{ecef} &= \left(\frac{a}{\sqrt{1 - e^2 \cdot \sin^2 p_{lat}}} + p_{alt} \right) \cdot \cos p_{lat} \cdot \sin p_{lon}, \\ z_{ecef} &= \left(\frac{a \cdot (1 - e^2)}{\sqrt{1 - e^2 \cdot \sin^2 p_{lat}}} + p_{alt} \right) \cdot \sin p_{lat}, \end{aligned} \tag{A.18}$$

where a is first eccentricity squared assigned by WGS-84, e^2 is the semi-major axis assigned by WGS-84. The next step is to transform ECEF positions into ENU position, if a position in ECEF ${}^{ECEF}\mathbf{p_o}$ is defined as the origin coordinate of ENU frame. The conversion expression is

$$\begin{bmatrix} x_{enu} \\ y_{enu} \\ z_{enu} \end{bmatrix} = \begin{bmatrix} -\sin p_{lon} & \cos p_{lon} & 0 \\ -\sin \phi_r \cdot \cos p_{lon} & -\sin \phi_r \cdot \sin p_{lon} & \cos \phi_r \\ \cos \phi_r \cdot \cos p_{lon} & -\cos \phi_r \cdot \sin p_{lon} & \sin \phi_r \end{bmatrix} \cdot \begin{bmatrix} x_{ecef} - \mathbf{p_{o,x}} \\ y_{ecef} - \mathbf{p_{o,y}} \\ z_{ecef} - \mathbf{p_{o,z}} \end{bmatrix}, \tag{A.19}$$

where $\phi_r = \arctan \frac{z_{ecef}}{\sqrt{(\mathbf{p_{o,x}})^2 + (\mathbf{p_{o,y}})^2}}$.

A.5 Usage of Coordinate Mapping in Localization

In the following chapters, the transformations among these three spaces are required. Specifically, given the coordinate of a point in one set, it is necessary to determine where it will appear in the other two coordinates. The mapping relationships are illustrated in Fig. A.3. The transformation functions between perspective space and BEV space are f_{IPM} and f_{IPM}^{-1}, while f_{ENU}, and f_{ENU}^{-1} are introduced to transfer pixel positions between BEV space and ENU space. Besides, the mapping calculations from geodetic coordinate to ENU coordinate are also demonstrated.

Lane marking detection methods and vehicle localization technologies are realized in the three spaces: perspective space, BEV space and ENU space. In lane marking detection part, the input image is transformed from perspective space to BEV space. Image processing, model estimation and self-assessment indicator are also achieved in BEV space. The output lane markings are transmitted to perspective space for vision

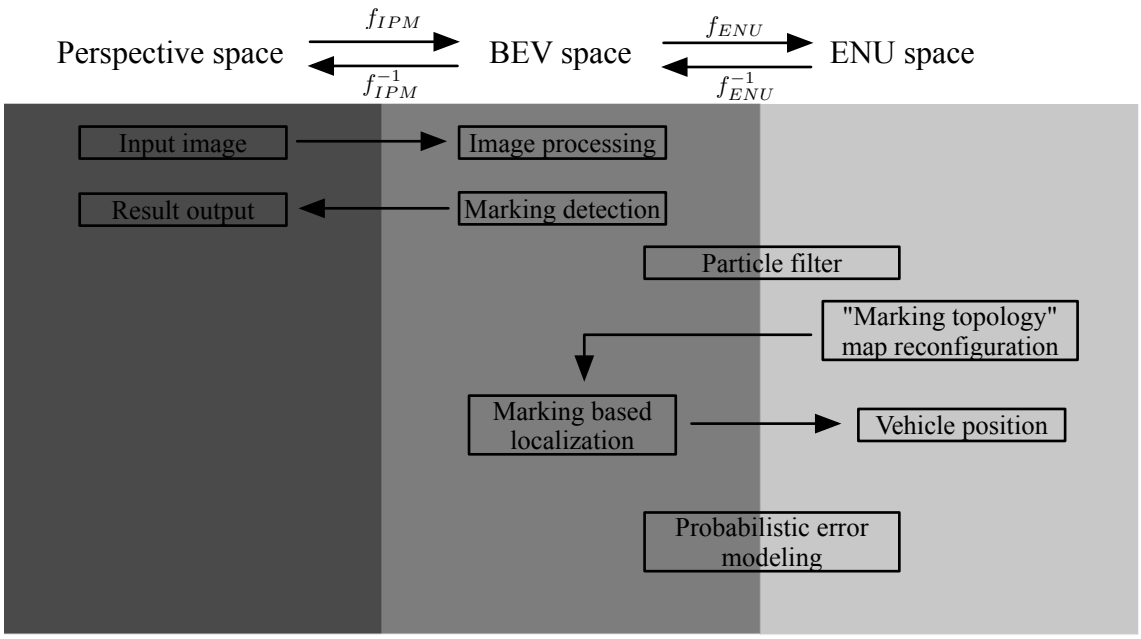


Figure A.3: *Project structure in different spaces. Different steps in the proposed method require different spaces.*

show. In vehicle localization part, particle filter predicts vehicle motion in BEV space, and then transfer the relative movements into ENU space. "Marking topology" from map database is reconfigured in ENU space. And these markings are transformed to BEV space in MKE based localization part. The improved vehicle position is outputted into ENU space. Finally, the potential errors are detecting in both BEV and ENU spaces.

Appendix B

Lane Marking Pixels Selection

Case 1: 1-lane Segment

Let $S_l^{bev} = S_{cur,l}^{bev} \cup S_{nxt,l}^{bev}$ and $S_r^{bev} = S_{cur,r}^{bev} \cup S_{nxt,r}^{bev}$ the sets of left and right marking pixels, $S_{cur,l}^{bev}$ and $S_{cur,r}^{bev}$ mean the marking pixels of current cell, and $S_{nxt,l}^{bev}$ and $S_{nxt,r}^{bev}$ are the marking pixels of the following corresponding cell. All the possible cases are considered to enumerate the relationship between the current cell and the required front cell.

The first case is that the current segment $seg_{i,j}$ is 1-lane segment, as shown in Fig. B.1. So the marking pixels of current cell is expressed as

$$S_{cur,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,l}\left(x_{i,j,A}^{cell,l1} \leq x \leq x_{i,j,A}^{cell,l2}\right)\right\}, \tag{B.1}$$

$$S_{cur,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,r}\left(x_{i,j,A}^{cell,r1} \leq x \leq x_{i,j,A}^{cell,r2}\right)\right\}. \tag{B.2}$$

The next segment $seg_{i,j+1}$ can be 1-lane segment (Fig. B.1a), 2-lane segment (Fig. B.1b), and 3-lane segment (Fig. B.1c). In Fig. B.1a, the vehicle drives from a one lane road to another one lane road, only one possibility exists. So $S_{nxt,l}^{bev}$ and $S_{nxt,r}^{bev}$ are represented as:

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \tag{B.3}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \tag{B.4}$$

In Fig. B.1b, the vehicle drives from a 1-lane road to a 2-lane road, the vehicle trajectory is possible to forward from the single host lane to the left or right lane in front. So the possible left markings for the next cell can be the left marking of $cel_{i,j+1,A}$ and that of

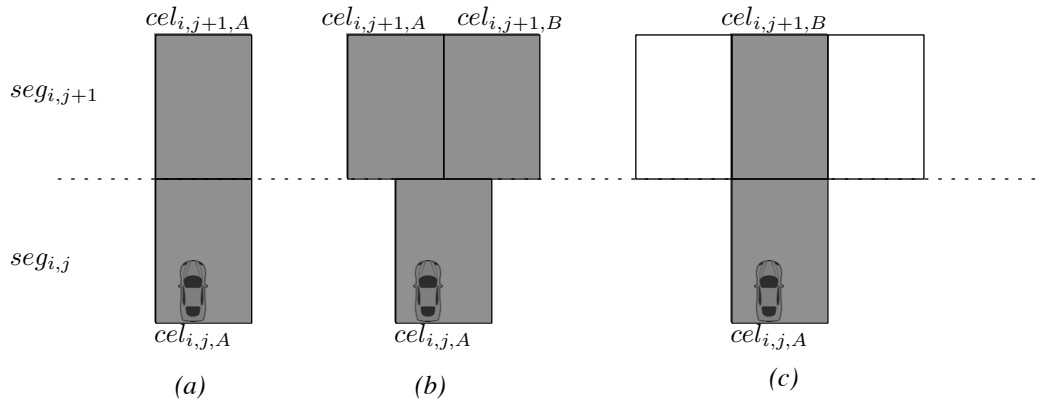


Figure B.1: *Corresponding cell relationship: 1-lane segment. (a) 1-lane road to a 1-lane road; (b) 1-lane road to a 2-lane road; (c) 1-lane road to a 3-lane road.*

of $cel_{i,j+1,B}$, while the potential right markings are the right marking of $cel_{i,j+1,A}$ and $cel_{i,j+1,B}$. $S^{bev}_{nxt,l}$ and $S^{bev}_{nxt,r}$ are represented as Eq. B.5 and Eq. B.6.

$$\begin{aligned} S^{bev}_{nxt,l} =& \{(x,y)\,|x,y\in\mathbb{Z}, y = k^{seg}_{i,j+1}\cdot x + b^{cel,l}_{i,j+1,A}\left(x^{cell,l1}_{i,j+1,A}\le x\le x^{cell,l2}_{i,j+1,A}\right), \\ & y = k^{seg}_{i,j+1}\cdot x + b^{cel,l}_{i,j+1,B}\left(x^{cell,l1}_{i,j+1,B}\le x\le x^{cell,l2}_{i,j+1,B}\right)\} \end{aligned} \tag{B.5}$$

$$\begin{aligned} S^{bev}_{nxt,r} =& \{(x,y)\,|x,y\in\mathbb{Z}, y = k^{seg}_{i,j+1}\cdot x + b^{cel,r}_{i,j+1,A}\left(x^{cell,r1}_{i,j+1,A}\le x\le x^{cell,r2}_{i,j+1,A}\right), \\ & y = k^{seg}_{i,j+1}\cdot x + b^{cel,r}_{i,j+1,B}\left(x^{cell,r1}_{i,j+1,B}\le x\le x^{cell,r2}_{i,j+1,B}\right)\}. \end{aligned} \tag{B.6}$$

In Fig. B.1(c), the vehicle drives from a 1-lane road to a 3-lane road, assume that there is no other entrances on the road, the road lane number is added from 1 lane to 3 lanes, so the most probable way is driving to the middle lane of the next segment. The possible markings for the next cell are marking of $cel_{i,j+1,B}$. $S^{bev}_{nxt,l}$ and $S^{bev}_{nxt,r}$ are represented as Eq. B.7 and Eq. B.18.

$$S^{bev}_{nxt,l} = \left\{(x,y)\,|x,y\in\mathbb{Z}, y = k^{seg}_{i,j+1}\cdot x + b^{cel,l}_{i,j+1,B}\left(x^{cell,l1}_{i,j+1,B}\le x\le x^{cell,l2}_{i,j+1,B}\right)\right\}, \tag{B.7}$$

$$S^{bev}_{nxt,r} = \left\{(x,y)\,|x,y\in\mathbb{Z}, y = k^{seg}_{i,j+1}\cdot x + b^{cel,r}_{i,j+1,B}\left(x^{cell,r1}_{i,j+1,B}\le x\le x^{cell,r2}_{i,j+1,B}\right)\right\}. \tag{B.8}$$

Case 2: 2-lane Segment

The second case is that the lane number of current segment $n^{seg,lane}_{i,j} = 2$, as Fig. B.2. Fig. B.2 lists all the possible driving route, from one gray cell in $seg_{i,j}$ to another gray

cell in $seg_{i,j+1}$. In Fig. B.2a to Fig. B.2c, the rover starts from the left cell $cel_{i,j,A}$, so $S^{bev}_{cur,l}$ and $S^{bev}_{cur,r}$ are represented as Eq. B.9 and Eq. B.10.

$$S^{bev}_{cur,l} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j} \cdot x + b^{cel,l}_{i,j,A}\left(x^{cell,l1}_{i,j,A} \leq x \leq x^{cell,l2}_{i,j,A}\right)\right\}, \tag{B.9}$$

$$S^{bev}_{cur,r} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j} \cdot x + b^{cel,r}_{i,j,A}\left(x^{cell,r1}_{i,j,A} \leq x \leq x^{cell,r2}_{i,j,A}\right)\right\}. \tag{B.10}$$

While in Fig. B.2d to Fig. B.2f, vehicle drives from the right cell $cel_{i,j,B}$. $S^{bev}_{cur,l}$ and $S^{bev}_{cur,r}$ are

$$S^{bev}_{cur,l} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j} \cdot x + b^{cel,l}_{i,j,B}\left(x^{cell,l1}_{i,j,B} \leq x \leq x^{cell,l2}_{i,j,B}\right)\right\}, \tag{B.11}$$

$$S^{bev}_{cur,r} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j} \cdot x + b^{cel,r}_{i,j,B}\left(x^{cell,r1}_{i,j,B} \leq x \leq x^{cell,r2}_{i,j,B}\right)\right\}. \tag{B.12}$$

For $S^{bev}_{nxt,l}$ and $S^{bev}_{nxt,r}$, Fig. B.2a to Fig. B.2f are analyzed respectively.

In Fig. B.2a,

$$S^{bev}_{nxt,l} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j+1} \cdot x + b^{cel,l}_{i,j+1,A}\left(x^{cell,l1}_{i,j+1,A} \leq x \leq x^{cell,l2}_{i,j+1,A}\right)\right\}, \tag{B.13}$$

$$S^{bev}_{nxt,r} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j+1} \cdot x + b^{cel,r}_{i,j+1,A}\left(x^{cell,r1}_{i,j+1,A} \leq x \leq x^{cell,r2}_{i,j+1,A}\right)\right\}. \tag{B.14}$$

In Fig. B.2b,

$$S^{bev}_{nxt,l} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j+1} \cdot x + b^{cel,l}_{i,j+1,A}\left(x^{cell,l1}_{i,j+1,A} \leq x \leq x^{cell,l2}_{i,j+1,A}\right)\right\}, \tag{B.15}$$

$$S^{bev}_{nxt,r} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j+1} \cdot x + b^{cel,r}_{i,j+1,A}\left(x^{cell,r1}_{i,j+1,A} \leq x \leq x^{cell,r2}_{i,j+1,A}\right)\right\}. \tag{B.16}$$

In Fig. B.2c,

$$\begin{aligned} S^{bev}_{nxt,l} = \{&(x,y)\,|x,y \in \mathbb{Z}, y = k^{seg}_{i,j+1} \cdot x + b^{cel,l}_{i,j+1,A}\left(x^{cell,l1}_{i,j+1,A} \leq x \leq x^{cell,l2}_{i,j+1,A}\right), \\ &y = k^{seg}_{i,j+1} \cdot x + b^{cel,l}_{i,j+1,B}\left(x^{cell,l1}_{i,j+1,B} \leq x \leq x^{cell,l2}_{i,j+1,B}\right)\} \end{aligned} \tag{B.17}$$

$$S_{nxt,r}^{bev} = \{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \le x \le x_{i,j+1,A}^{cell,r2}\right), \quad (B.18)$$
$$y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,B}^{cel,r}\left(x_{i,j+1,B}^{cell,r1} \le x \le x_{i,j+1,B}^{cell,r2}\right)\}$$

In Fig. B.2d,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \le x \le x_{i,j+1,A}^{cell,l2}\right)\right\}, \quad (B.19)$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \le x \le x_{i,j+1,A}^{cell,r2}\right)\right\}. \quad (B.20)$$

In Fig. B.2e,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,B}^{cel,l}\left(x_{i,j+1,B}^{cell,l1} \le x \le x_{i,j+1,B}^{cell,l2}\right)\right\}, \quad (B.21)$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,B}^{cel,r}\left(x_{i,j+1,B}^{cell,r1} \le x \le x_{i,j+1,B}^{cell,r2}\right)\right\}. \quad (B.22)$$

In Fig. B.2f,

$$S_{nxt,l}^{bev} = \{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,B}^{cel,l}\left(x_{i,j+1,B}^{cell,l1} \le x \le x_{i,j+1,B}^{cell,l2}\right), \quad (B.23)$$
$$y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,C}^{cel,l}\left(x_{i,j+1,C}^{cell,l1} \le x \le x_{i,j+1,C}^{cell,l2}\right)\},$$

$$S_{nxt,r}^{bev} = \{(x,y)\,|x,y\in\mathbb{Z}, y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,B}^{cel,r}\left(x_{i,j+1,B}^{cell,r1} \le x \le x_{i,j+1,B}^{cell,r2}\right), \quad (B.24)$$
$$y = k_{i,j+1}^{seg}\cdot x + b_{i,j+1,C}^{cel,r}\left(x_{i,j+1,C}^{cell,r1} \le x \le x_{i,j+1,C}^{cell,r2}\right)\}.$$

Case 3: 3-lane Segment

The third case is that the lane number of current segment $n_{i,j}^{seg,lane} = 3$, as Fig. B.3. Fig. B.3 lists all the possible driving route, from one gray cell in $seg_{i,j}$ to another gray cell in $seg_{i,j+1}$. In Fig. B.3a to Fig. B.3c, the rover starts from the left cell $cel_{i,j,A}$, so $S_{cur,l}^{bev}$ and $S_{cur,r}^{bev}$ are represented as Eq. B.25 and Eq. B.26.

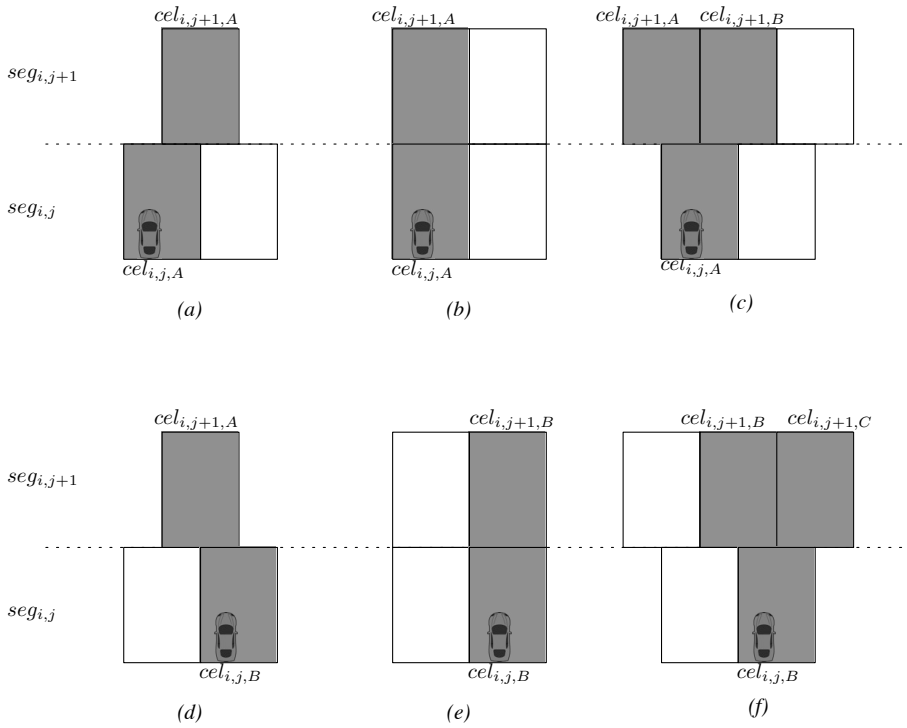


Figure B.2: *Corresponding cell relationship: 2-lane segment. (a), (d): 2-lane road to 1-lane road; (b), (e): 2-lane road to 2-lane road; (c), (f): 2-lane road to 3-lane road.*

$$S_{cur,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,l}\left(x_{i,j,A}^{cell,l1} \leq x \leq x_{i,j,A}^{cell,l2}\right)\right\}, \tag{B.25}$$

$$S_{cur,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,A}^{cel,r}\left(x_{i,j,A}^{cell,r1} \leq x \leq x_{i,j,A}^{cell,r2}\right)\right\}. \tag{B.26}$$

While in Fig. B.3d to Fig. B.3f, vehicle drives from the right cell $cel_{i,j,B}$. $S_{cur,l}^{bev}$ and $S_{cur,r}^{bev}$ are

$$S_{cur,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,B}^{cel,l}\left(x_{i,j,B}^{cell,l1} \leq x \leq x_{i,j,B}^{cell,l2}\right)\right\}, \tag{B.27}$$

$$S_{cur,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,B}^{cel,r}\left(x_{i,j,B}^{cell,r1} \leq x \leq x_{i,j,B}^{cell,r2}\right)\right\}. \tag{B.28}$$

In Fig. B.3g to Fig. B.3i, vehicle drives from the right cell $cel_{i,j,C}$. $S_{cur,l}^{bev}$ and $S_{cur,r}^{bev}$ are

$$S_{cur,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,C}^{cel,l}\left(x_{i,j,C}^{cell,l1} \leq x \leq x_{i,j,C}^{cell,l2}\right)\right\}, \tag{B.29}$$

$$S_{cur,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j}^{seg} \cdot x + b_{i,j,C}^{cel,r}\left(x_{i,j,C}^{cell,r1} \leq x \leq x_{i,j,C}^{cell,r2}\right)\right\}. \tag{B.30}$$

For $S_{nxt,l}^{bev}$ and $S_{nxt,r}^{bev}$, Fig. B.3a to Fig. B.3i are analyzed respectively.

In Fig. B.3a,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \quad \text{(B.31)}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \quad \text{(B.32)}$$

In Fig. B.3b,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \quad \text{(B.33)}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \quad \text{(B.34)}$$

In Fig. B.3c,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \quad \text{(B.35)}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \quad \text{(B.36)}$$

In Fig. B.3d,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \quad \text{(B.37)}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \quad \text{(B.38)}$$

In Fig. B.3e,

$$\begin{aligned} S_{nxt,l}^{bev} = &\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right), \quad \text{(B.39)} \\ &y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,B}^{cel,l}\left(x_{i,j+1,B}^{cell,l1} \leq x \leq x_{i,j+1,B}^{cell,l2}\right)\}, \end{aligned}$$

$$S_{nxt,r}^{bev} = \{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right), \quad y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,B}^{cel,r}\left(x_{i,j+1,B}^{cell,r1} \leq x \leq x_{i,j+1,B}^{cell,r2}\right)\}. \tag{B.40}$$

In Fig. B.3f,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,B}^{cel,l}\left(x_{i,j+1,B}^{cell,l1} \leq x \leq x_{i,j+1,B}^{cell,l2}\right)\right\}, \tag{B.41}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,B}^{cel,r}\left(x_{i,j+1,B}^{cell,r1} \leq x \leq x_{i,j+1,B}^{cell,r2}\right)\right\}. \tag{B.42}$$

In Fig. B.3g,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,l}\left(x_{i,j+1,A}^{cell,l1} \leq x \leq x_{i,j+1,A}^{cell,l2}\right)\right\}, \tag{B.43}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,A}^{cel,r}\left(x_{i,j+1,A}^{cell,r1} \leq x \leq x_{i,j+1,A}^{cell,r2}\right)\right\}. \tag{B.44}$$

In Fig. B.3h,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,B}^{cel,l}\left(x_{i,j+1,B}^{cell,l1} \leq x \leq x_{i,j+1,B}^{cell,l2}\right)\right\}, \tag{B.45}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,B}^{cel,r}\left(x_{i,j+1,B}^{cell,r1} \leq x \leq x_{i,j+1,B}^{cell,r2}\right)\right\}. \tag{B.46}$$

In Fig. B.3i,

$$S_{nxt,l}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,C}^{cel,l}\left(x_{i,j+1,C}^{cell,l1} \leq x \leq x_{i,j+1,C}^{cell,l2}\right)\right\}, \tag{B.47}$$

$$S_{nxt,r}^{bev} = \left\{(x,y)\,|x,y \in \mathbb{Z}, y = k_{i,j+1}^{seg} \cdot x + b_{i,j+1,C}^{cel,r}\left(x_{i,j+1,C}^{cell,r1} \leq x \leq x_{i,j+1,C}^{cell,r2}\right)\right\}. \tag{B.48}$$

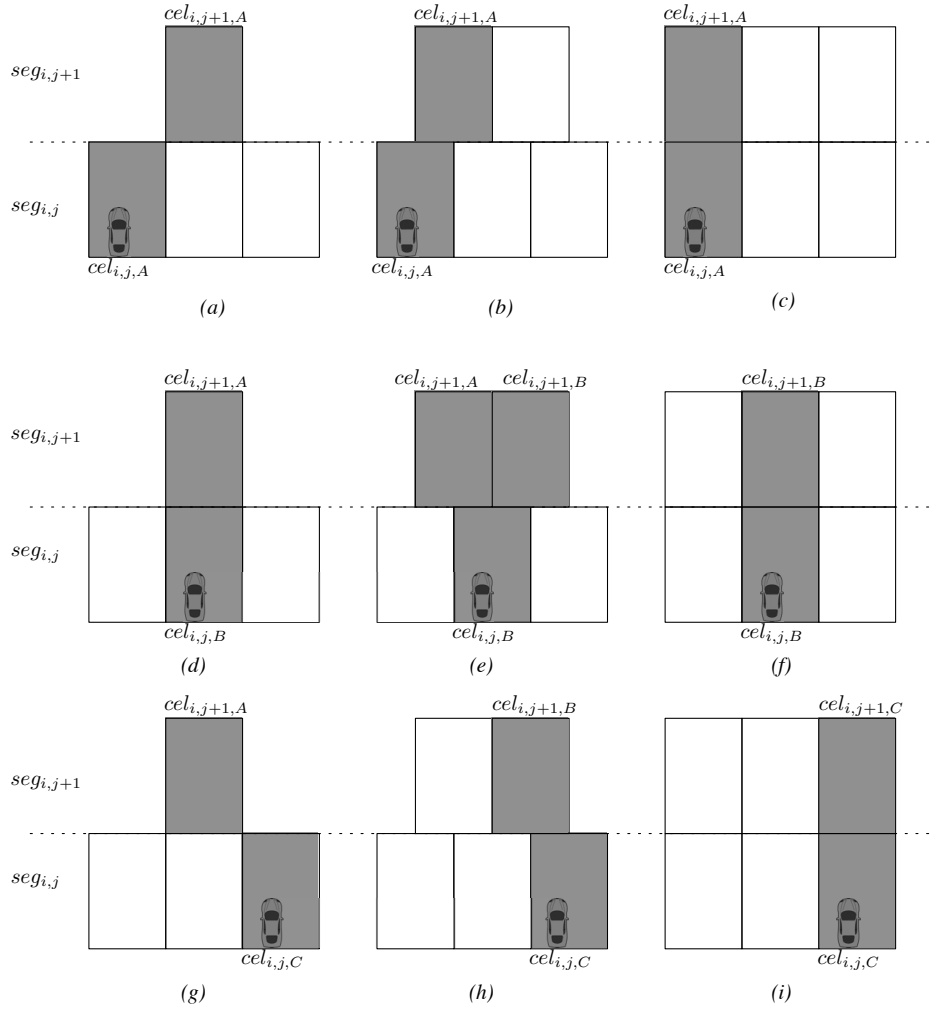


Figure B.3: *Corresponding cell relationship: 3-lane segment. 2-lane segment. (a), (d), (g): 3-lane road to 1-lane road; (b), (e), (h): 3-lane road to 2-lane road; (c), (f), (i): 3-lane road to 3-lane road.*

References

J. M. Álvarez, A. M. López, and R. Baldrich, "Shadow resistant road segmentation from a mobile monocular system," in *Pattern Recognition and Image Analysis*. Springer, 2007, pp. 9–16.

M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 7–12.

A. Amini, R. M. Vaghefi, J. M. de la Garza, and R. M. Buehrer, "Improving gps-based vehicle positioning for intelligent transportation systems," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 1023–1029.

B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Dover Publications, 2012.

N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2003, pp. 558–563.

A. Assidiq, O. Khalifa, R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *Conf. Computer and Communication Engineering*, 2008, pp. 82–88.

A. Ather, "A quality analysis of openstreetmap data," Master's thesis, University College London, 2009.

L. Bai, Y. Wang, and M. Fairhurst, "An extended hyperbola model for road tracking for video-based personal navigation," *Knowledge-Based Systems*, vol. 21, no. 3, pp. 265–272, 2008.

F. S. Barickman, L. Smith, and R. Jones, "Lane departure warning system research and test development," in *Proc. Conf. on the Enhanced Safety of Vehicles*, no. 07-0495, 2007.

A. Barth, J. Siegemund, and J. Schwehr, "Fast and precise localization at stop intersections," in *IEEE Intelligent Vehicles Symposium Workshops*, 2013, pp. 75–80.

M. Bellino, Y. L. De Meneses, P. Ryser, and J. Jacot, "Lane detection algorithm for an onboard camera," in *European Workshop on Photonics in the Automobile*, 2005, pp. 102–111.

N. Benmansour, R. Labayrade, D. Aubert, S. Glaser, and D. Gruyer, "A model driven 3d lane detection system using stereovision," in *IEEE Conf. Control Automation Robotic and Vision (ICARCV)*, 2008, pp. 1277–1282.

J. Bentley and H. Maurer, "Efficient worst-case data structures for range searching," *Acta Informatica*, vol. 13, pp. 155–168, 1980.

D. Bernstein and A. Kornhauser, "An introduction to map matching for personal navigation assistant," *Transportation Research Part C: Emerging Technologies*, vol. 1, pp. 91–108, 2000.

M. Bertozzi and A. Broggi, "Gold: a parallel real-time stereo vision system for generic obstacle and lane detection," in *IEEE Trans. on Image Processing*, 1998, pp. 62–81.

A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and kalman filter." in *Proc. IEEE Conf. Image Processing (ICIP)*, 2009, pp. 3261–3264.

A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti, "A layered approach to robust lane detection at night," in *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, 2009, pp. 51–57.

A. Broggi and S. Cattani, "An agent based evolutionary approach to path detection for off-road vehicle guidance," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1164–1173, 2006.

R. S. Bucy and K. D. Senne, "Digital synthesis of non-linear filters," *Automatica*, vol. 7, no. 3, pp. 287–298, 1971.

G. Burzio, L. Guidotti, G. Perboli, M. Settanni, R. Tadei, and F. Tesauri, "Investigating the impact of a lane departure warning system in real driving conditions-a subjective field operational test," in *European Conference on Human Centred Design for Intelligent Transport Systems*, 2010, pp. 313–322.

W. Chen, M. Yu, Z. Li, and Y. Chen, "Integrated vehicle navigation system for urban applications," in *Proc. Conf. on Global Navigation Satellite Systems (GNSS)*, 2003, pp. 15–22.

H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 571–582, 2006.

S. Chindaro, K. Sirlantzis, and F. Deravi, "Colour space fusion for texture recognition," in *EURASIP Conference focused on Video, Image Processing and Multimedia Communications*, vol. 1, 2003, pp. 181–186.

D. Cui, J. Xue, S. Du, and N. Zheng, "Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration," in *IEEE/RSJ Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 4958–4964.

R. Dahyot, "Statistical hough transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1502–1509, 2009.

N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

R. Danescu and S. Nedevschi, "Probabilistic lane tracking in difficult road scenarios using stereovision," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 272–282, 2009.

R. Danescu, S. Nedevschi, and T.-B. To, "A stereovision-based lane detector for marked and non-marked urban roads," in *IEEE Conf. Intelligent Computer Communication and Processing*, 2007, pp. 81–88.

N. El-Sheimy and X. Niu, "The promise of mems to the navigation community," *Inside GNSS*, vol. 2, no. 2, pp. 46–56, 2007.

P. Enge, T. Walter, S. Pullen, C. Kee, Y.-C. Chao, and Y.-J. Tsai, "Wide area augmentation of the global positioning system," *Proceedings of the IEEE*, vol. 84, no. 8, pp. 1063–1088, 1996.

M. Enzweiler, P. Greiner, C. Knoppel, and U. Franke, "Towards multi-cue urban curb recognition," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 902–907.

J. O. Esparza-Jimenez, M. Devy, and J. Gordillo, "Ekf-based slam fusing heterogeneous landmarks," in *IEEE Conf. Information Fusion (FUSION)*, 2014, pp. 1–8.

T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, September 2013.

N. Goodman, "Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction)," *Annals of mathematical statistics*, pp. 152–177, 1963.

N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEEE Proc. Radar and Signal Processing*, vol. 140, no. 2, 1993, pp. 107–113.

M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007.

D. Grimmer, *Finding straight edges in radar images using deformable templates*. University of Michigan–Dearborn, 1993.

D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2014, pp. 674–680.

M. Haklay, "How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets," *Environment and Planning B: Planning and Design*, vol. 37, pp. 682–703, 2010.

J. Handschin, "Monte carlo techniques for prediction and filtering of non-linear stochastic processes," *Automatica*, vol. 6, no. 4, pp. 555–563, 1970.

J. Handschin and D. Q. Mayne, *International journal of control*, vol. 9, no. 5, pp. 547–559, 1969.

R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Trans. on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 309–318, 2004.

J. Hernández, B. Marcotegui *et al.*, "Filtering of artifacts and pavement segmentation from mobile lidar data," in *ISPRS Workshop on Laser Scanning*, 2009.

A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

P. V. Hough, "Machine analysis of bubble chamber pictures," in *Conf. High Energy Accelerators and Instrumentation*, vol. 73, 1959.

M. Hu, W. Yang, M. Ren, and J. Yang, "A vision based road detection algorithm," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, vol. 2, 2004, pp. 846–850.

A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and lidar," *Autonomous Robots*, vol. 26, no. 2-3, pp. 103–122, 2009.

J. Huang and H.-S. Tan, "A low-order dgps-based vehicle positioning system under urban environment," *IEEE/ASME Trans. on Mechatronics*, vol. 11, no. 5, pp. 567–575, 2006.

S.-S. Huang, C.-J. Chen, P.-Y. Hsiao, and L.-C. Fu, "On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, vol. 3, 2004, pp. 2456–2461.

P. J. Huber, *Robust statistics*. Springer, 2011.

J. Hur, S.-N. Kang, and S.-W. Seo, "Multi-lane detection in urban driving environments using conditional random fields," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1297–1302.

J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer vision, graphics, and image processing*, vol. 44, no. 1, pp. 87–116, 1988.

M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 1996, pp. 343–356.

K. Ishikawa, K. Kobayashi, and K. Watanabe, "A lane detection method for intelligent ground vehicle competition," in *Proc. SICE Annual Conference*, vol. 1, 2003, pp. 1086–1089.

F. Janda, S. Pangerl, E. Lang, and E. Fuchs, "Road boundary detection for run-off road prevention based on the fusion of video and radar," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1173–1178.

M. Javadi, M. Hannan, S. Samad, and A. Hussain, "A robust vision-based lane boundaries detection approach for intelligent vehicles," *Information Technology Journal*, vol. 11, no. 9, pp. 1184–1192, 2012.

A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Dover Publications, 2007.

R. Jiang, R. Klette, T. Vaudrey, and S. Wang, "New lane model and distance transform for lane detection and tracking," in *Computer Analysis of Images and Patterns*. Springer, 2009, pp. 1044–1052.

R. Jiang, R. Klette, T. Vaudery, and S. Wang, "Lane detection and tracking using a new lane model and distance transform," *Machine Vision and Applications*, vol. 22, no. 4, pp. 721–737, June 2011.

Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," in *IEEE Conf. Image Analysis and Signal Processing (IASP)*, 2010, pp. 114–117.

R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Fluids Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 1137–1142.

D. J. Kang, J. W. Choi, and I. S. Kweon, "Finding and tracking road lanes using line-snakes," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 1996, pp. 189–194.

S.-N. Kang, S. Lee, J. Hur, and S.-W. Seo, "Multi-lane detection based on accurate geometric lane estimation in highway scenarios," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 221–226.

I. Katramados, S. Crumpler, and T. P. Breckon, "Real-time traversable surface detection by colour space fusion and temporal analysis," in *Computer Vision Systems*. Springer, 2009, pp. 265–274.

B. Kim and K. Yi, "Probabilistic states prediction algorithm using multi-sensor fusion and application to smart cruise control systems," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 888–895.

Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. on Intelligent Transportation Systems*, pp. 16–26, March 2008.

G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.

H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 96–103.

A. L. Kornhauser, "Darpa urban challenge princeton university technical paper," Princeton University, Tech. Rep., 2007.

R. Labayrade, J. Douret, J. Laneurit, and R. Chapuis, "A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety

assistance," *IEICE Transactions on Information and Systems*, vol. 89, no. 7, pp. 2092–2100, 2006.

A. H. Lai and N. H. Yung, "Lane detection by orientation and length discrimination," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 30, no. 4, pp. 539–548, 2000.

H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 719–724.

H. Li and F. Nashashibi, "Robust real-time lane detection based on lane mark segment features and general a priori knowledge," in *IEEE Conf. Robotics and Biomimetics (ROBIO)*, 2011, pp. 812–817.

Q. Li, N. Zheng, and H. Cheng, "Lane boundary detection using an adaptive randomized hough transform," in *Proc. the 5th World Congress on Intelligent Control and Automation (WCICA)*, vol. 5, 2004, pp. 4084–4088.

S. Li and Y. Shimomura, "Lane marking detection by side fisheye camera," in *IEEE/RSJ Conf. Intelligent Robots and Systems (IROS)*, September 2008, pp. 606–611.

X. Li, E. Seignez, and P. Loonis, "Reliability-based driver drowsiness detection using dempster-shafer theory," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, December 2012, pp. 1059–1064.

X. Li, E. Seignez, A. Lambert, and P. Loonis, "Real-time driver drowsiness estimation by multi-source information fusion with dempster-shafer theory," *Transactions of the Institute of Measurement and Control*, 2013.

A. Linarth and E. Angelopoulou, "On feature templates for particle filter based lane detection," in *IEEE Conf. Intelligent Transportation Systems (ITSC)*, October 2011, pp. 1721–1726.

C. Lipski, B. Scholz, K. Berger, C. Linz, T. Stich, and M. Magnor, "A fast and robust approach to lane marking detection and lane tracking," in *Southwest Symposium on Image Analysis and Interpretation*, 2008, pp. 57–60.

G. Liu, F. Worgotter, and I. Markelic, "Lane shape estimation using a partitioned particle filter for autonomous driving," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, May 2011, pp. 1627–1633.

J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113–119, 1996.

J. S. Liu and R. Chen, "Blind deconvolution via sequential imputations," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, 1995.

W. Liu and S. Li, "An effective lane detection algorithm for structured road in urban," in *Intelligent Science and Intelligent Data Engineering*. Springer, 2013, pp. 759–767.

X. Liu, Q. Song, and P. Li, "A parabolic detection algorithm based on kernel density estimation," in *Emerging Intelligent Computing Technology and Applications*, 2009, pp. 405–412.

P. Lombardi, M. Zanin, and S. Messelodi, "Switching models for vision-based on-board road detection," in *IEEE Conf. Intelligent Transportation Systems (ITSC)*, September 2005, pp. 67–72.

W. Lu, S. A. Rodríguez F., E. Seignez, and R. Reynaud, "Probabilistic error model for a lane marking based vehicle localization coupled to open source maps," in *IEEE Conf. Intelligent Transportation Systems (ITSC)*, 2014, pp. 360–365.

W. Lu, S. A. Rodríguez F., E. Seignez, and R. Reynuad, "Monocular multi-kernel based lane marking detection," in *Proc. IEEE Conf. Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, June 2014.

W. Lu, Y. Zheng, Y. Q. Ma, and T. Liu, "An integrated approach to recognition of lane marking and road boundary," in *Workshop on Knowledge Discovery and Data Mining*, 2008, pp. 649–653.

W. Lu, S. A. Rodríguez F., E. Seignez, and R. Reynaud, "An improved approach for vision-based lane marking detection and tracking," in *International Conference on Electrical, Control and Automation Engineering*, December 2013, pp. 382–386.

W. Lu, E. Seignez, S. A. Rodríguez F., and R. Reynaud, "Lane marking based vehicle localization using particle filter and multi-kernel estimation," in *IEEE Conf. Control Automation Robotic and Vision (ICARCV), Conference Best Paper Award*, 2014.

E. Malis and E. Marchand, "Experiments with robust estimation techniques in real-time robot vision," in *IEEE/RSJ Conf. Intelligent Robots and Systems (IROS)*, 2006, pp. 223–228.

J. Marais, M. Berbineau, and M. Heddebaut, "Land mobile gnss availability and multipath evaluation tool," *IEEE Trans. on Vehicular Technology*, vol. 54, no. 5, pp. 1697–1704, 2005.

J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, March 2006.

Y. Meng, "Improved positioning of land vehicle in its using digital map and other accessory information," Ph.D. dissertation, Hong Kong Polytechnic University, 2006.

S. Miura and S. Kamijo, "Gps error correction by multipath adaptation," *International Journal of Intelligent Transportation Systems Research*, January 2014.

M. E. E. Najjar and P. Bonnifait, "A road-matching method for precise vehicle localization using belief theory and kalman filtering," *Autonomous Robots*, vol. 19, no. 2, pp. 173–191, 2005.

R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, vol. 11, no. 2, pp. 125–139, 2001.

S. Nedevschi, F. Oniga, R. Danescu, T. Graf, and R. Schmidt, "Increased accuracy stereo approach for 3d lane detection," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2006, pp. 42–49.

A. V. Nefian and G. R. Bradski, "Detection of drivable corridors for off-road autonomous navigation," in *Image Processing, 2006 IEEE International Conference on*, 2006, pp. 3025–3028.

A. Nemra and N. Aouf, "Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering," *Sensors Journal, IEEE*, vol. 10, no. 4, pp. 789–798, 2010.

M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2007, pp. 315–320.

M. Nieto, L. Salgado, F. Jaureguizar, and J. Arróspide, "Robust multiple lane road modeling based on perspective analysis," in *Proc. IEEE Conf. Image Processing (ICIP)*, 2008, pp. 2396–2399.

M. Nieto, J. A. Laborda, and L. Salgadoi, "Road environment modeling using robust perspective analysis and recursive bayesian segmentation," *Machine Vision and Applications*, vol. 22, no. 6, pp. 927–945, November 2011.

M. Nieto, A. Cortés, O. Otaegui, J. Arróspide, and L. Salgado, "Real-time lane tracking using rao-blackwellized particle filter," *Journal of Real-Time Image Processing*, pp. 1–13, 2012.

D. Obradovic, H. Lenz, and M. Schupfner, "Fusion of map and sensor data in a modern car navigation system," *Journal of VSLI Signal Processing*, vol. 45, no. 1-2, pp. 111–122, 2006.

W. Y. Ochieng, M. A. Quddus, and R. B. Noland, "Map-matching in complex urban road networks," *Brazilian Society of Cartography, Geodesy, Photogrammetry and Remote Sensing (SBC)*, 2003.

W. Y. Ochieng, M. Quddus, and R. B. Noland, "Map-matching in complex urban road networks," *Revista Brasileira de Cartografia*, vol. 2, no. 55, 2009.

T. Ogawa and K. Takagi, "Lane recognition using on-vehicle lidar," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2006, pp. 540–545.

E. Orhan, "Particle filtering," University of Rochester, Tech. Rep., 2012.

User manual: covers RT2000, RT3000 and RT4000 products, 140611th ed., Oxford Technical Solutions Limited, 2014.

B. P. Phuyal, "Method and use of aggregated dead reckoning sensor and gps data for map matching," in *Proc. Conf. the Institute of Navigation (ION)*, September 2002, pp. 430–437.

V. Pradeep, G. Medioni, and J. Weiland, "Piecewise planar modeling for step detection using stereo vision," in *Workshop on Computer Vision Applications for the Visually Impaired*, September 2008.

B. Qin, Z. Chong, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Curb-intersection feature based monte carlo localization on urban roads," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, 2012, pp. 2640–2646.

M. Quddus, W. Ochieng, and R. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 5, pp. 312–328, October 2007.

M. A. Quddus, R. B. Nolanda, and W. Y. Ochienga, "A high accuracy fuzzy logic based map matching algorithm for road transport," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 10, no. 3, pp. 103–115, 2006.

M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Integrity of map-matching algorithms," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 4, pp. 283–302, 2006.

K. Redmill, T. Kitajima, and U. Ozguner, "Dgps/ins integrated positioning for control of automated vehicle," in *IEEE Conf. Intelligent Transportation Systems (ITSC)*, 2001, pp. 172–178.

S. A. Rodríguez F., V. Fremont, P. Bonnifait, and V. Cherfaoui, "An embedded multi-modal system for object localization and tracking," *IEEE Trans. on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 42–53, November 2012.

S. A. Rodríguez F., V. Fremont, P. Bonnifait, and V. Cherfaoui, "Multi-modal object detection and localization for high integrity driving assistance," *Machine Vision Applications*, vol. 25, no. 3, pp. 583–598, April 2014.

N. B. Romdhane, M. Hammami, and H. Ben-Abdallah, "A comparative study of vision-based lane detection methods," in *Advanced Concepts for Intelligent Vision Systems*, vol. 6915, 2011, pp. 46–57.

C. Rotaru, T. Graf, and J. Zhang, "Extracting road features from color images using a cognitive approach," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2004, pp. 298–303.

F. Samadzadegan, A. Sarafraz, and M. Tabibi, "Automatic lane detection in image sequences for vision-based navigation purposes," *ISPRS Image Engineering and Vision Metrology*, 2006.

H. Sawano and M. Okada, "A road extraction method by an active contour model with inertia and differential features," *IEICE Transactions on Information and Systems*, vol. 89, no. 7, pp. 2257–2267, 2006.

M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 449–454.

F. Schule, R. Schweiger, and K. Dietmayer, "Augmenting night vision video images with longer distance road course information," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1233–1238.

S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, "Robust lane detection in urban environments," in *IEEE/RSJ Conf. Intelligent Robots and Systems (IROS)*, 2007, pp. 123–128.

A. Seibert, M. Hahnel, A. Tewes, and R. Rojas, "Camera based detection and classification of soft shoulders, curbs and guardrails," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 853–858.

E. Seignez, M. Kieffer, A. Lambert, E. Walter, and T. Maurin, "Real-time bounded-error state estimation for vehicle tracking," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 34–48, January 2009.

W. Seo, S. Hwang, J. Park, and J.-M. Lee, "Precise outdoor localization with a gps–ins integration system," *Robotica*, vol. 31, no. 03, pp. 371–379, 2013.

Y.-W. Seo and R. R. Rajkumar, "Utilizing instantaneous driving direction for enhancing lane-marking detection," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 170–175.

X. Shi, B. Kong, and F. Zheng, "A new lane detection method based on feature pattern," in *Conf. Image and Signal Processing (CISP)*, 2009, pp. 1–5.

N. Shimomura, K. Fujimoto, T. Oki, and H. Muro, "An algorithm for distinguishing the types of objects on the road using laser radar and vision," *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 189–195, 2002.

S. Sivaraman and M. M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 906–917, 2013.

I. Skog and P. Handel, "In-car positioning and navigation technologies—a survey," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4–21, March 2009.

Y. S. Son, S.-H. Lee, and C. C. Chung, "Predictive virtual lane using relative motions between a vehicle and lanes," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 771–776.

N. Soquet, D. Aubert, and N. Hautiere, "Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2007, pp. 160–165.

H. W. Sorenson and D. L. Alspach, "Recursive bayesian estimation using gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

T.-Y. Sun, S.-J. Tsai, and V. Chan, "Hsi color model based lane-marking detection," in *IEEE Conf. Intelligent Transportation Systems (ITSC)*, 2006, pp. 1168–1172.

N. Sunderhauf, M. Obst, S. Lange, G. Wanielik, and P. Protzel, "Switchable constraints and incremental smoothing for online mitigation of non-line-of-sight and multipath effects," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 262–268.

C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, no. 6, pp. 2520–2530, 2005.

Z. Tao, P. Bonnifait, and V. Frémont, "Lane marking aided vehicle localization," in *IEEE Conf. Intelligent Transportation Systems (ITSC)*, November 2013.

Z. Tao, P. Bonnifait, V. Frémont, and J. Ibañez-Guzman, "Mapping and localization using gps, lane markings and proprioceptive," in *IEEE/RSJ Conf. Intelligent Robots and Systems (IROS)*, November 2013, pp. 406–412.

J.-P. Tarel, S.-S. Ieng, and P. Charbonnier, "Using robust estimation algorithms for tracking explicit curves," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2002, pp. 492–507.

Q.-B. Truong and B.-R. Lee, "New lane detection algorithm for autonomous vehicles using computer vision," in *International Conf. Control, Automation and Systems (ICCAS)*, 2008, pp. 1208–1213.

Q. B. Truong, B. R. Lee, N. G. Heo, Y. J. Yum, and J. G. Kim, "Lane boundaries detection algorithm using vector lane concept," in *Proc. IEEE Conf. Robotics and Automation (ICRA)*, 2008, pp. 2319–2325.

S. Tuohy, D. O'Cualain, E. Jones, and M. Glavin, "Distance determination for an automobile environment using inverse perspective mapping in opencv," in *Irish Signals and Systems Conference*, 2010.

C. Wang, Z. Hu, and R. Chapuis, "Predictive lane detection by interaction with digital road map," *Information and Media Technologies*, vol. 7, no. 1, pp. 383–392, 2012.

C.-C. Wang, S.-S. Huang, and L.-C. Fu, "Driver assistance system for lane detection and vehicle recognition with night vision," in *IEEE/RSJ Conf. Intelligent Robots and Systems (IROS)*, 2005, pp. 3530–3535.

Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Trans. on Intelligent Transportation Systems*, vol. 9, no. 4, pp. 570–579, 2008.

Y. Wang, E. K. Teoha, and D. Shen, "Lane detection and tracking using b-snake," *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, April 2004.

Q. Wen, Z. Yang, Y. Song, and P. Jia, "Road boundary detection in complex urban environment based on low-resolution vision," in *11th Joint International Conference on Information Sciences*, 2008.

M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*. Springer New York, 1997.

C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1-6, pp. 91–108, February–December 2000.

M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-grade global positioning system (gps) accuracy and reliability," *Journal of Forestry*, vol. 103, no. 4, pp. 169–173, 2005.

S.-J. Wu, H.-H. Chiang, J.-W. Perng, C.-J. Chen, B.-F. Wu, and T.-T. Lee, "The heterogeneous systems integration design and implementation for lane keeping on a vehicle," *IEEE Trans. on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 246–263, 2008.

T. Wu and A. Ranganathan, "Vehicle localization using road markings," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 1185–1190.

K. Yamaguchi, A. Watanabe, T. Naito, and Y. Ninomiya, "Road region estimation using a sequence of monocular images," in *International Conf. Pattern Recognition (ICPR)*, 2008, pp. 1–4.

S. Yenikaya, G. Yenikaya, and E. Düven, "Keeping the vehicle on the road: A survey on on-road lane detection systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, October 2013.

Y. U. Yim and S.-Y. Oh, "Three-feature based automatic lane detection algorithm (tfalda) for autonomous driving," *IEEE Trans. on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 219–225, 2003.

B. Yu, W. Zhang, and Y. Cai, "A lane departure warning system based on machine vision," in *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA)*, vol. 1, 2008, pp. 197–201.

F. Zhang, H. Stahle, C. Chen, C. Buckl, and A. Knoll, "A lane marking extraction approach based on random finite set statistics," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1143–1148.

W. Zhu, F. Liu, Z. Li, X. Wang, and S. Zhang, "A vision based lane detection and tracking algorithm in automatic drive," in *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA)*, December 2008, pp. 799–803.

C. Zinoune, P. Bonnifait, and J. Ibañez-Guzmán, "Integrity monitoring of navigation systems using repetitive journeys," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2014, pp. 274–280.